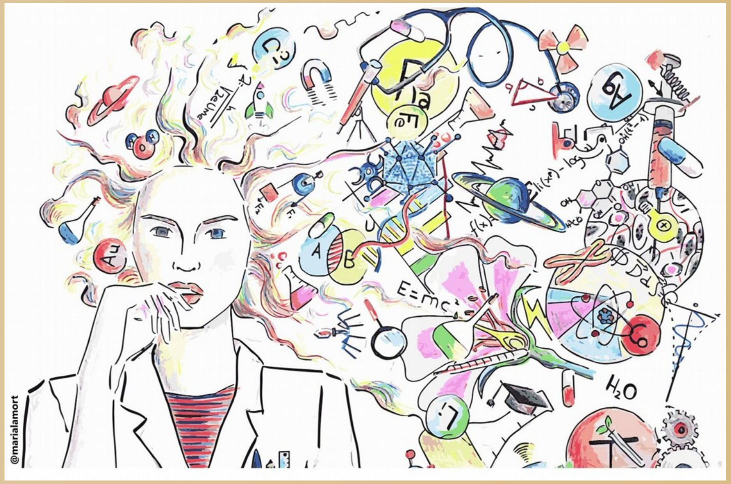
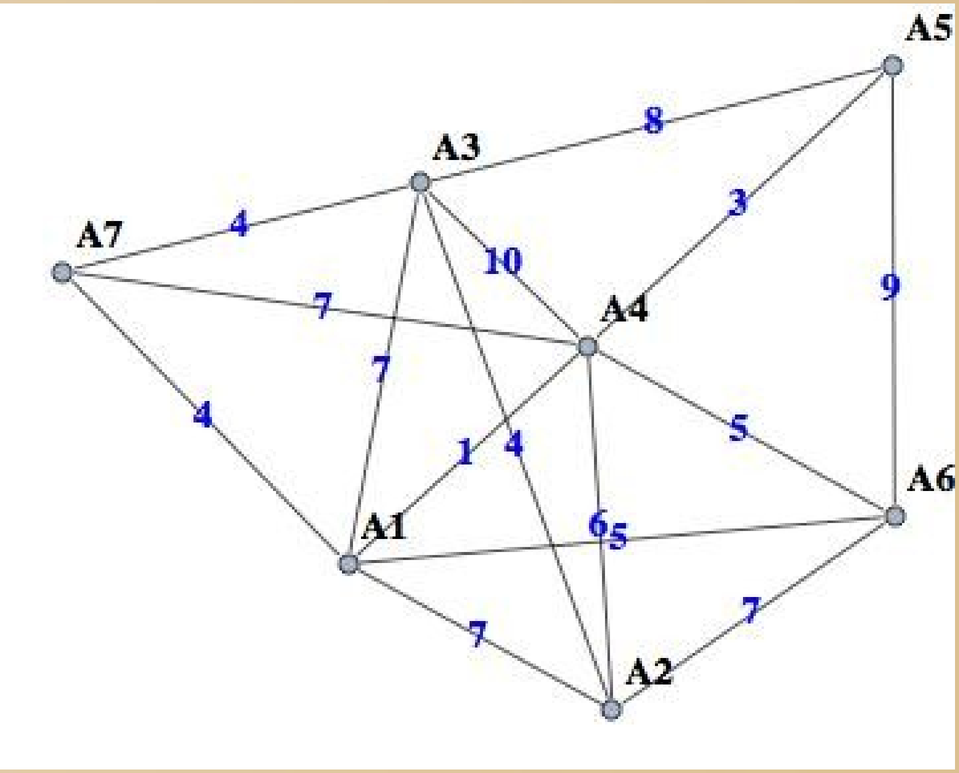
Problemes de Viatjants

Carlos D'Andrea

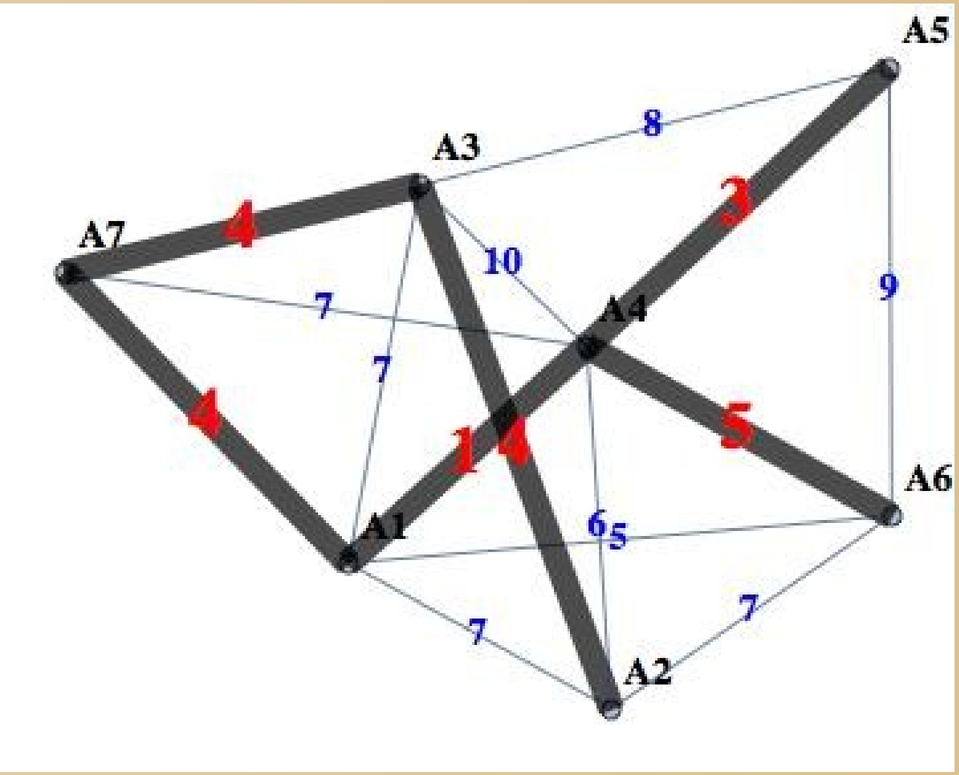


UNIVERSITAT DE BARCELONA

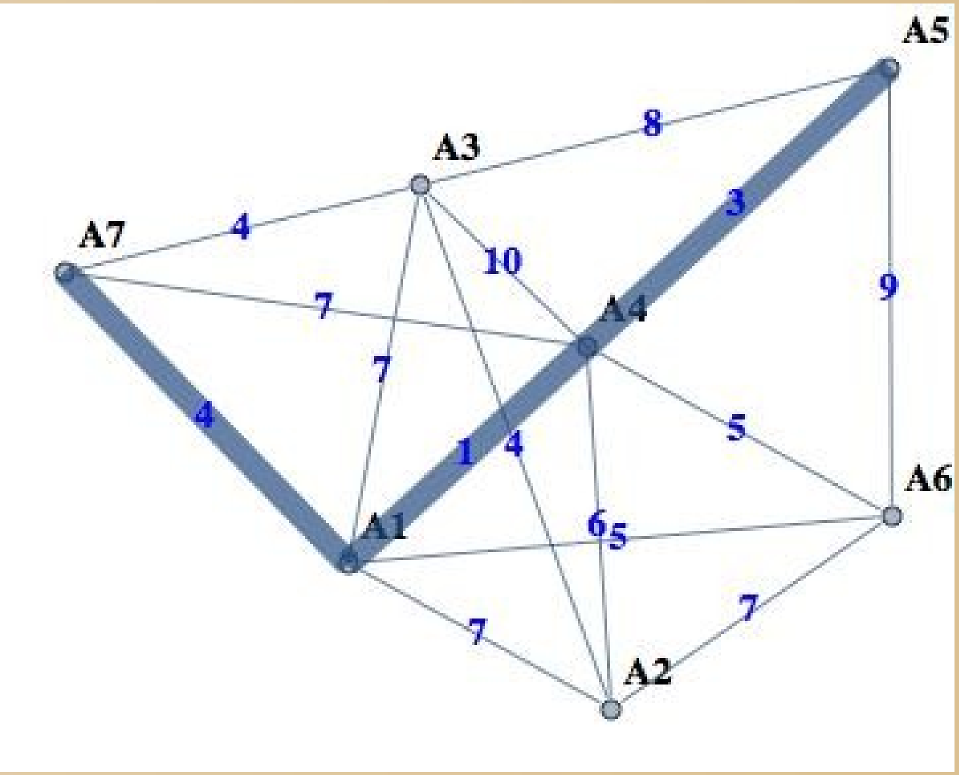
Problemas “de caminos”



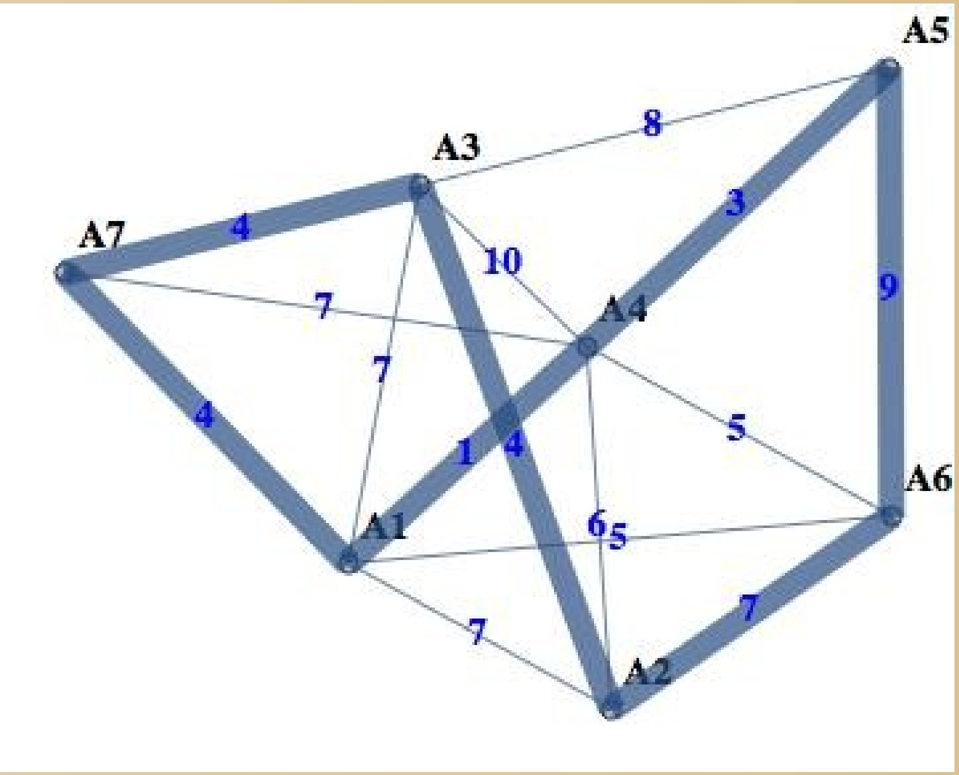
El Conector Mínimo



El Camino más Corto



El Problema del Viajante



¿Cómo se resuelven?

Se podrían calcular ***todas*** *las posibles opciones y quedarse con la más “corta”*

¿Cómo se resuelven?

Se podrían calcular ***todas*** *las posibles opciones y quedarse con la más "corta"*

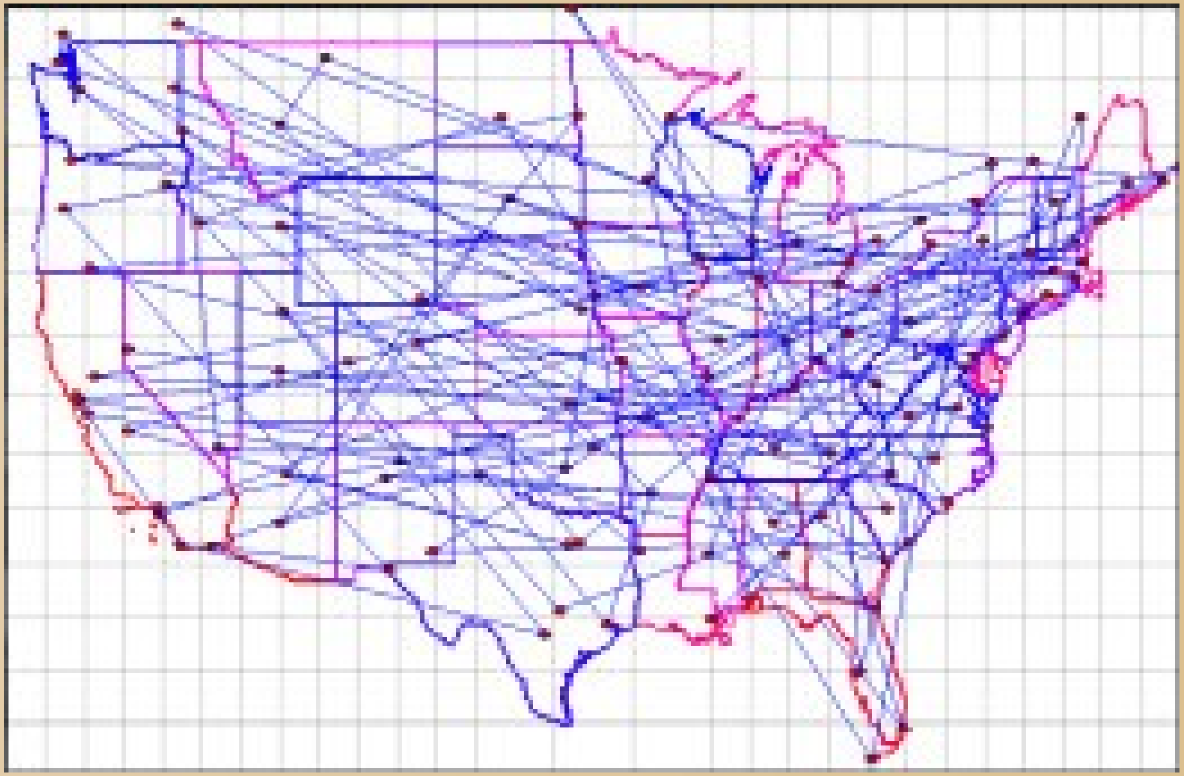
Hay 15 "longitudes", 2^{15}= 32.768 posibilidades

¿Cómo se resuelven?

*Se podrían calcular **todas** las posibles opciones y quedarse con la más "corta"*

Hay 15 "longitudes", 2^{15}= 32.768 posibilidades

¡Y podrían ser más!!



Es fácil para un ordenador?

Es fácil para un ordenador?

49 ciudades ***1950***

Es fácil para un ordenador?

49 ciudades ***1950***

2392 ciudades ***1980***

Es fácil para un ordenador?

49 ciudades ***1950***

2392 ciudades ***1980***

85900 ciudades ***2006***

Es fácil para un ordenador?

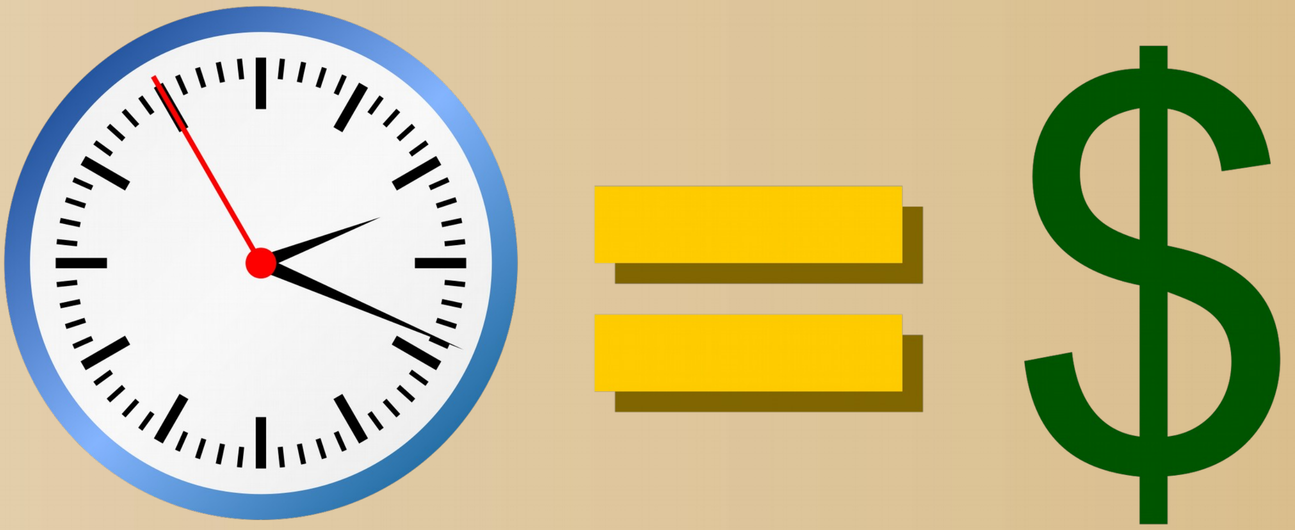
49 ciudades ***1950***

2392 ciudades ***1980***

85900 ciudades ***2006***

.... Luego de 280 días de cálculo!

Time is Money!



¿Cómo calcular“rápido”?

El idioma del ordenador

Un ***algoritmo*** *es como una* **receta de cocina**

El idioma del ordenador

*Un **algoritmo** es como una **receta de cocina***
*un **conjunto de instrucciones** que se sigue para resolver un problema*

Algoritmo eficiente

Los algoritmos eficientes son aquelllos que un ordenador puede realizar en tiempo razonable

¿"Razonable"?

Si la cantidad de pasos a realizar es **exponencial** (como el factorial del número de vértices), el algoritmo **NO** es eficiente

Test de rapidez

$3^{16}=3x3x3x3x3x3x3x3x3x3x3x3x3x3x3x3=43046721$

Test de rapidez

$3^{16}=3x3x3x3x3x3x3x3x3x3x3x3x3x3x3x3=43046721$

15 operaciones

Test de rapidez

$3^{16}=3x3x3x3x3x3x3x3x3x3x3x3x3x3x3x3=43046721$

15 operaciones

$3^2=9 \rightarrow 9^2=81 \rightarrow 81^2=6561 \rightarrow 6561^2=43046721$

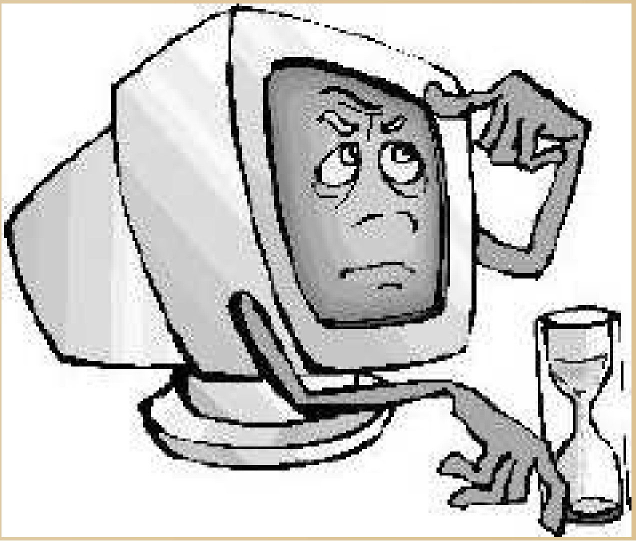
Test de rapidez

$3^{16}=3x3x3x3x3x3x3x3x3x3x3x3x3x3x3x3=43046721$

15 operaciones

$3^2=9 \rightarrow 9^2=81 \rightarrow 81^2=6561 \rightarrow 6561^2=43046721$

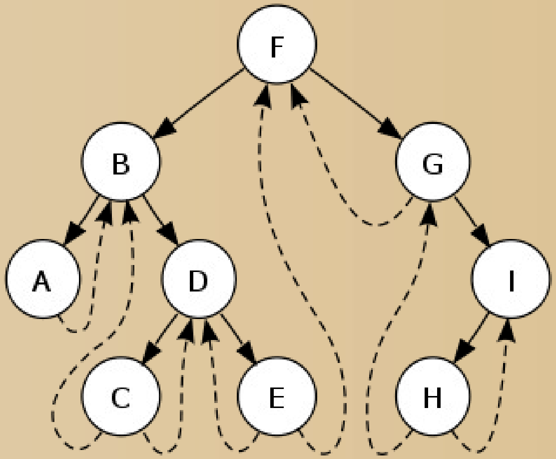
4 operaciones



Iterativo vs recursivo

$3^{2^{\wedge}n}$=3x3x3x3x3x......x3x3x3

2^n-1 productos ¡Exponencial!



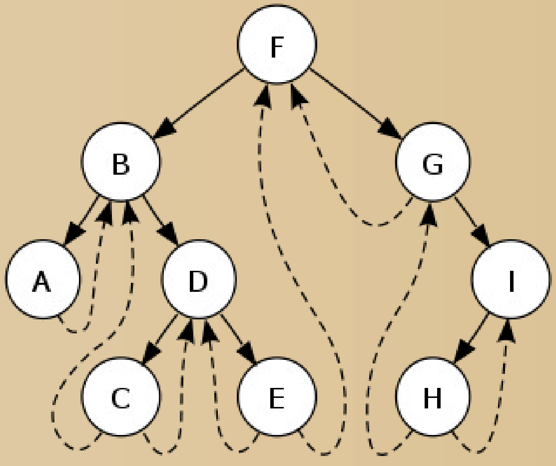
Iterativo vs recursivo

$3^{2^n}=3x3x3x3x3x......x3x3x3$

2^n-1 productos *¡Exponencial!*

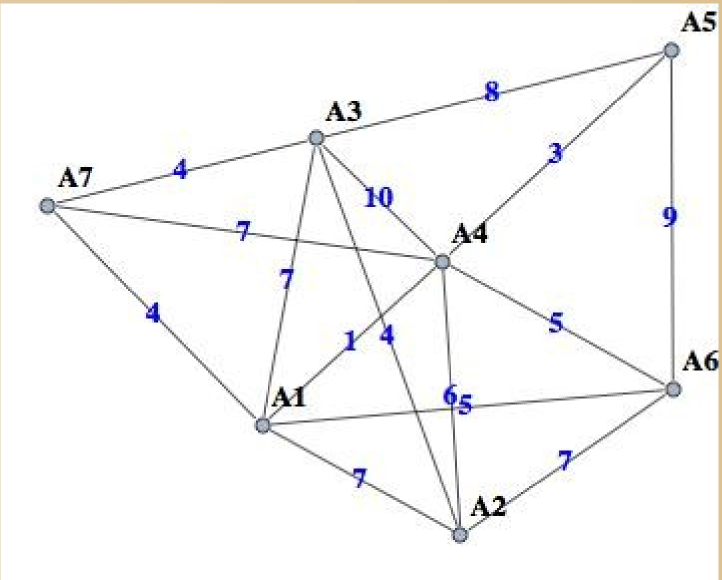
$3^2=9 \rightarrow 9^2=81 \rightarrow ..... \rightarrow (3^{2^{(n-1)}})^2$

n productos *¡Lineal!*



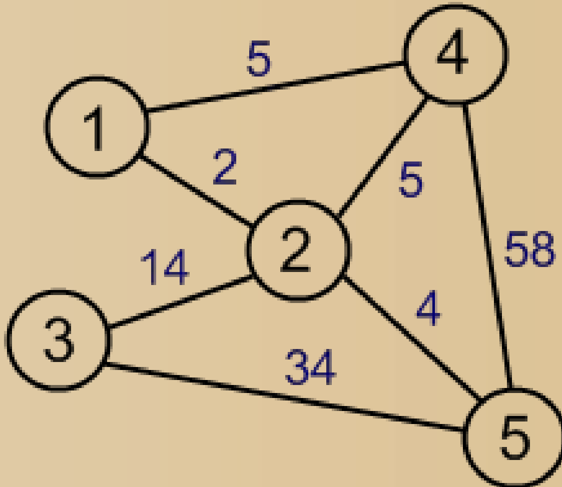
En nuestro caso...

¿Es posible encontrar algoritmos (recursivos, iterativos o...) que resuelvan los tres problemas sin tener que hacer una cantidad exponencial de operaciones?



Un poco de lenguaje

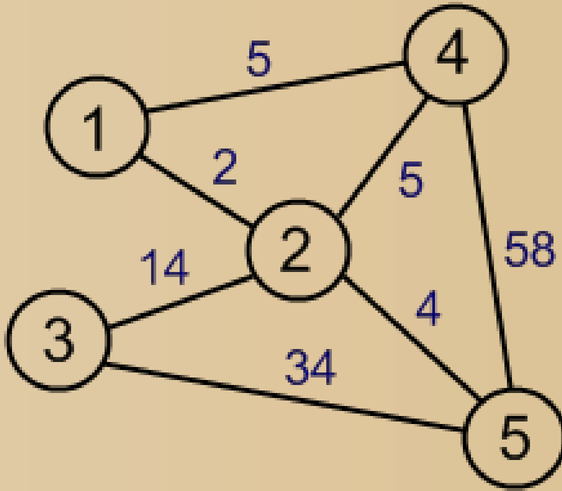
Un **grafo conexo** tiene



Un poco de lenguaje

Un **grafo conexo** tiene

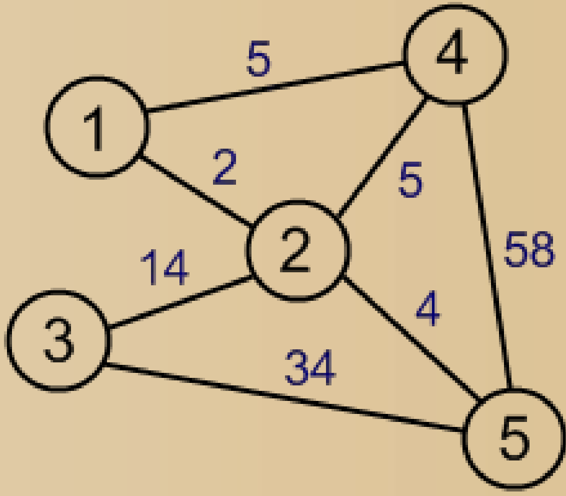
- **vértices** o **nodos**



Un poco de lenguaje

Un **grafo conexo** tiene

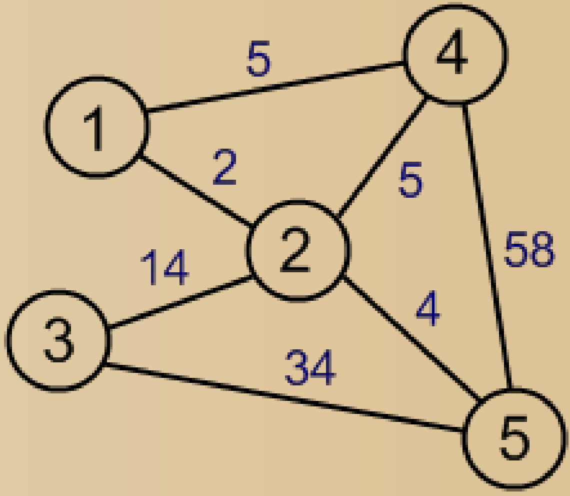
- **vértices** o **nodos**
- **aristas** o **lados**



Un poco de lenguaje

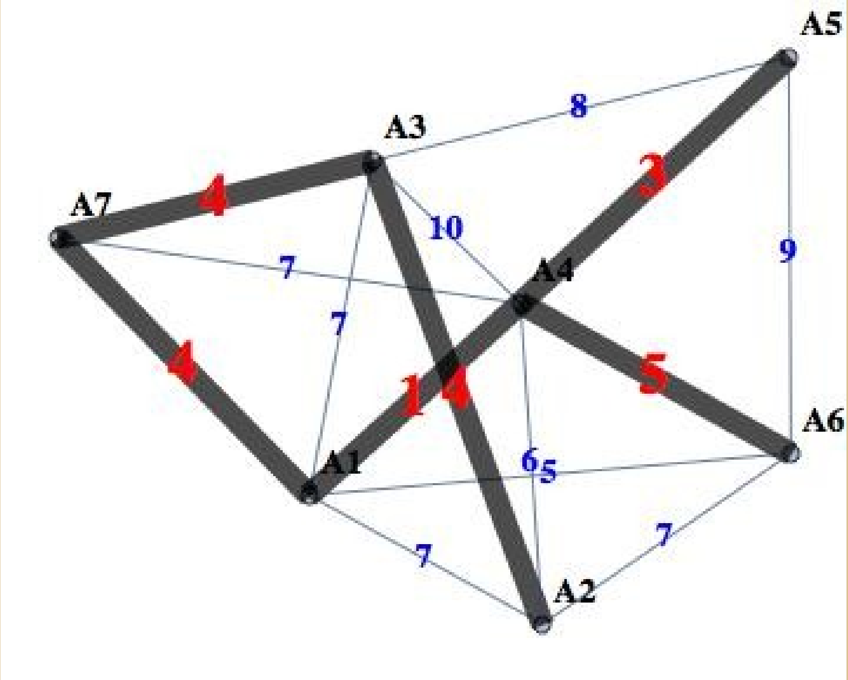
Un **grafo conexo** tiene

- **vértices** o **nodos**
- **aristas** o **lados**
- las aristas tienen **pesos** o **longitudes**



Conector Mínimo
(**Algoritmo de Prim**)

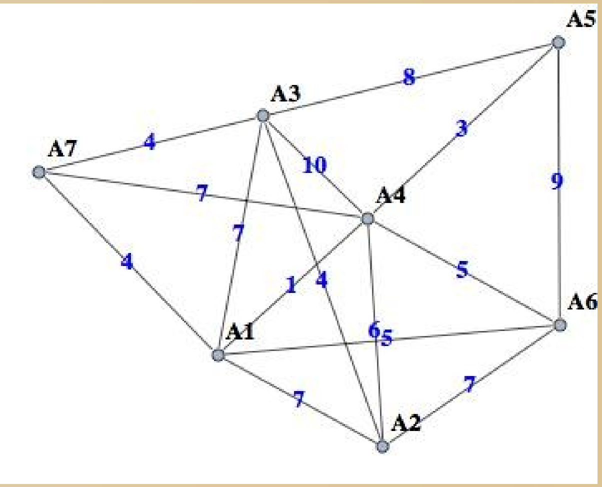
Dado un **grafo conexo**, encuentra una **red mínima** o un **conector mínimo** entre todos sus vértices



Algoritmo de Prim: paso 0

- Elegimos un vértice cualquiera **(A_1),** y lo "separamos" de los otros

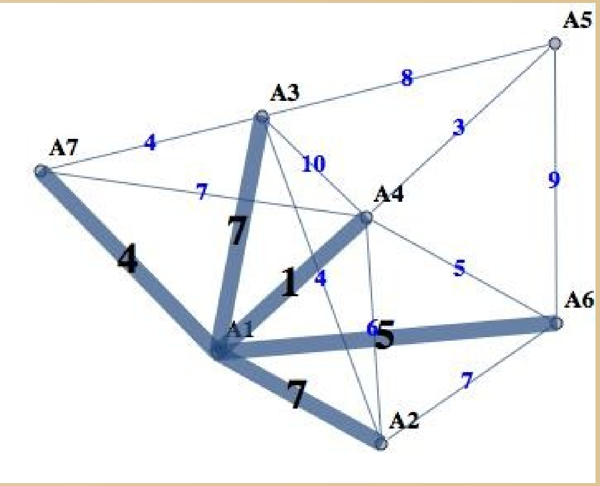
$V=\{A_1\}$ **$W=\{A_2,A_3,A_4,A_5,A_6,A_7\}$**



Algoritmo de Prim: 1º paso

- Calculamos todas las longitudes de aristas que salen de **V** y llegan a **W**

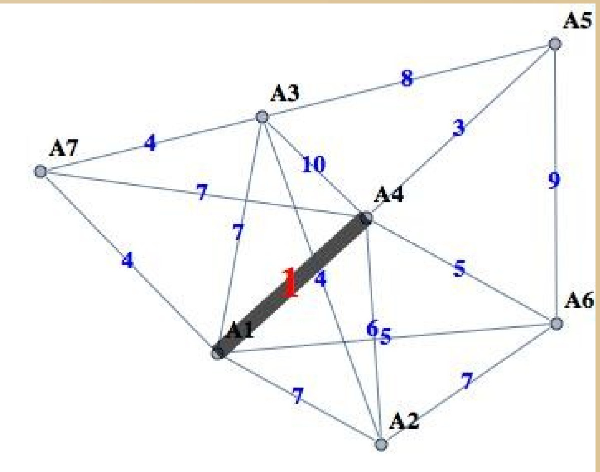
$V=\{A_1\}$ **$W=\{A_2,A_3,A_4,A_5,A_6,A_7\}$**



Algoritmo de Prim: 2º paso

➢ Nos quedamos con ***el*** lado de longitud más corta, agregamos el nuevo vértice a **V**, y lo quitamos de **W**

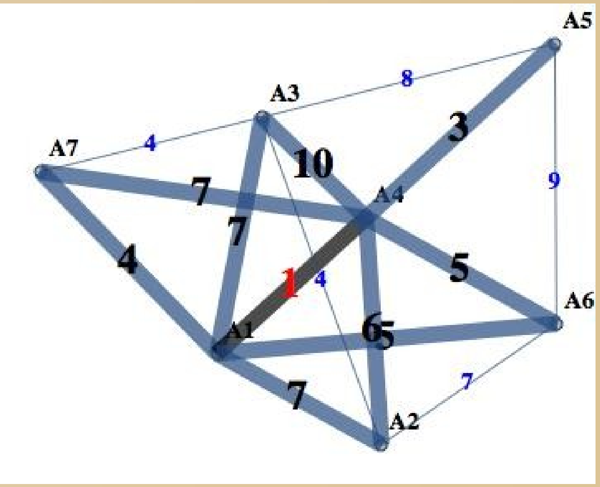
$V=\{A_1,A_4\}$ **$W=\{A_2,A_3,A_5,A_6,A_7\}$**



Algoritmo de Prim: 3º paso

- Como en el 1er paso, volvemos a calcular todas las longitudes de lados que conectan puntos de **V** con puntos de **W**

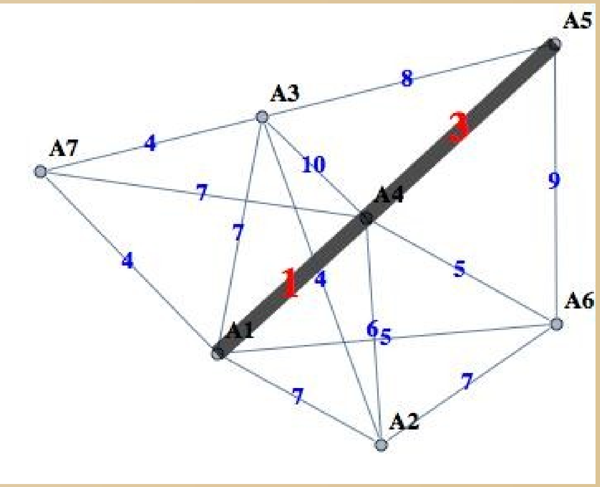
$V=\{A_1,A_4\}$ **$W=\{A_2,A_3,A_5,A_6,A_7\}$**



Algoritmo de Prim: 4º paso

- Como en el 2do paso, nos quedamos con ***el*** lado de menor longitud. Modificamos los conjuntos **V** y **W**

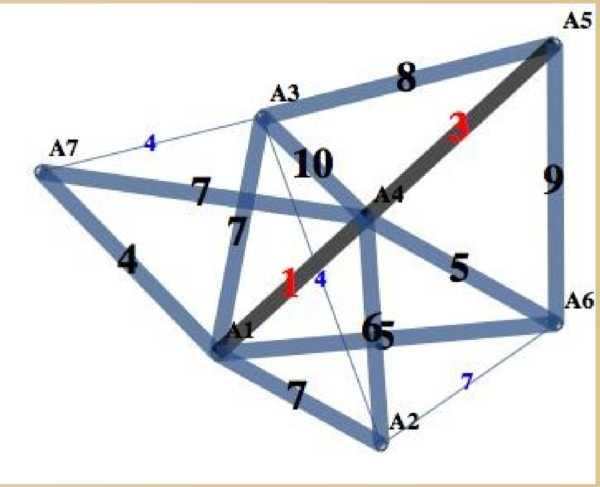
$V=\{A_1,A_4,A_5\}$ **$W=\{A_2,A_3,A_6,A_7\}$**



Algoritmo de Prim: 5º paso

- Repetimos el 1er paso

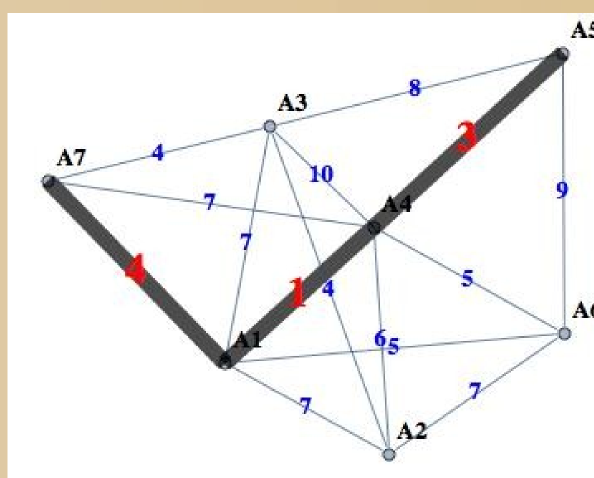
$V=\{A_1,A_4,A_5\}$ **$W=\{A_2,A_3,A_6,A_7\}$**



Algoritmo de Prim: 6º paso

- Repetimos el 2do paso

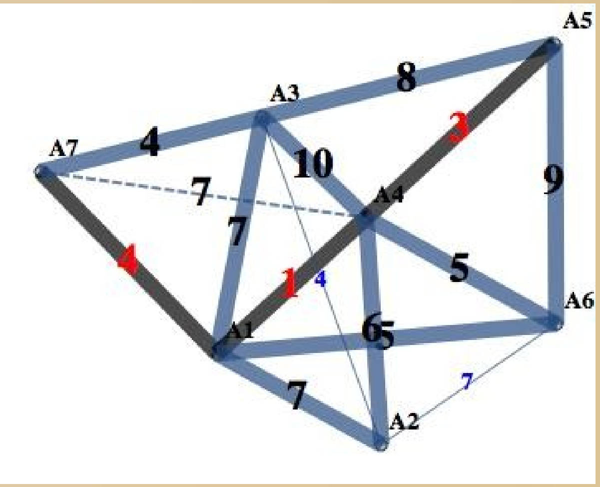
$V=\{A_1,A_4,A_5,A_7\}$ **$W=\{A_2,A_3,A_6\}$**



Algoritmo de Prim: 7º paso

- Repetimos el 1er paso

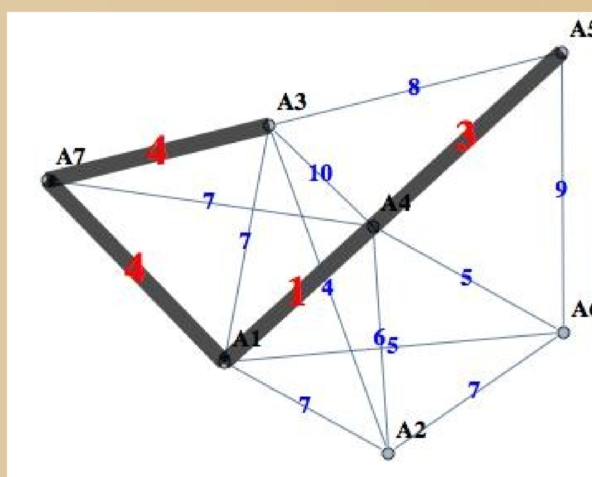
$V=\{A_1,A_4,A_5,A_7\}$ **$W=\{A_2,A_3,A_6\}$**



Algoritmo de Prim: 8º paso

- Repetimos el 2do paso

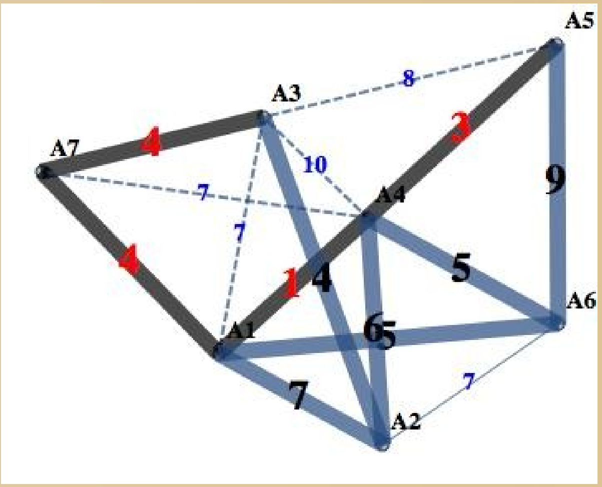
$V=\{A_1,A_4,A_5,A_7,A_3\}$ **$W=\{A_2,A_6\}$**



Algoritmo de Prim: 9º paso

- Repetimos el 1er paso

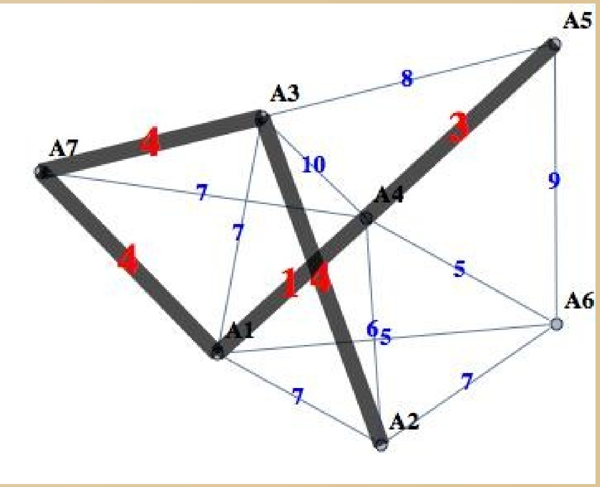
$V=\{A_1,A_4,A_5,A_7,A_3\}$ $W=\{A_2,A_6\}$



Algoritmo de Prim: 10º paso

- Repetimos el 2do paso

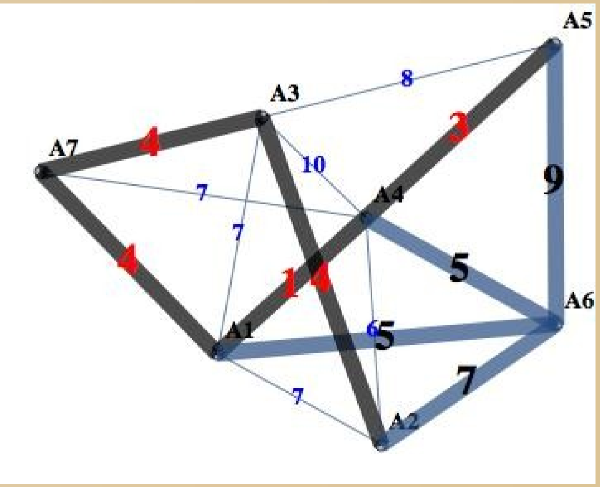
$V=\{A_1,A_4,A_5,A_7,A_3,A_2\}$ **$W=\{A_6\}$**



Algoritmo de Prim: finale

- Repetimos el 1er paso

$V=\{A_1,A_4,A_5,A_7,A_3,A_2\}$ $W=\{A_6\}$

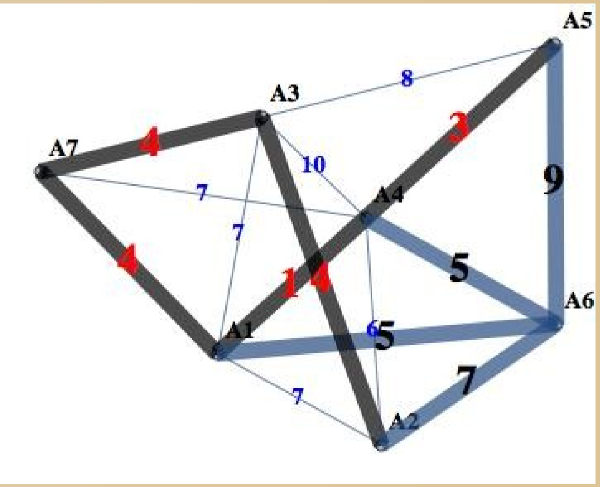


Algoritmo de Prim: finale

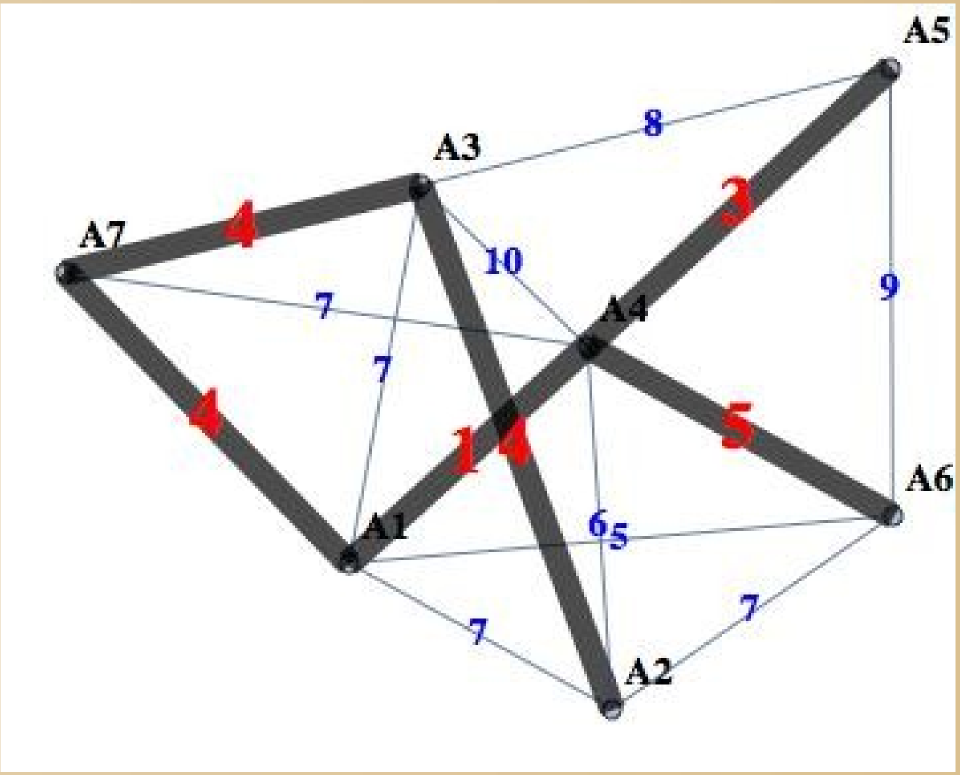
- Repetimos el 1er paso

  $V=\{A_1,A_4,A_5,A_7,A_3,A_2\}$ **$W=\{A_6\}$**

- y el 2do una vez más

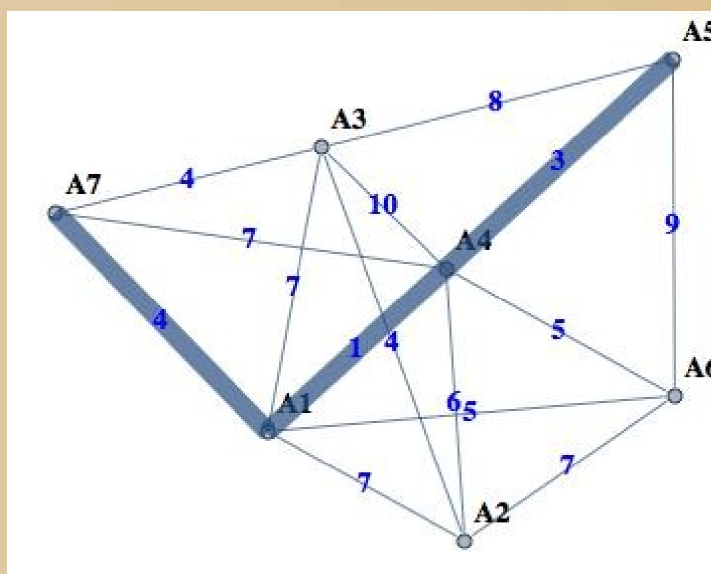


Y *la* red mínima es...



El Camino más corto
(**Algoritmo de Dijkstra**)

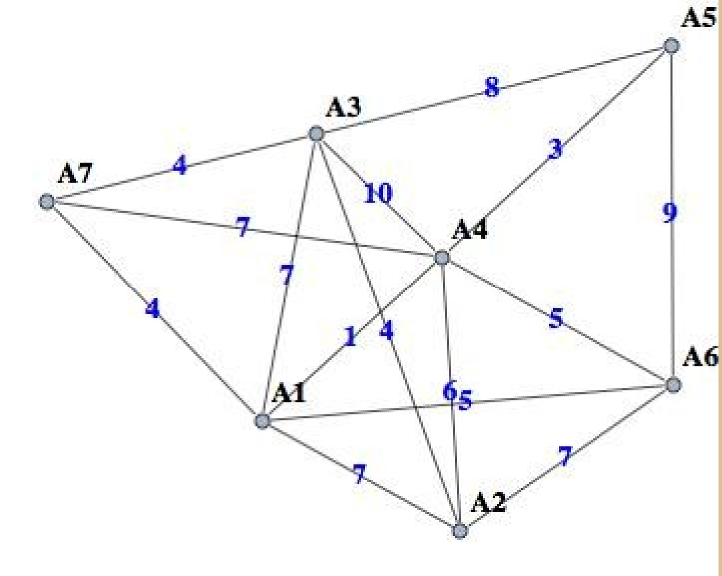
Dados un **grafo conexo** y dos de sus **vértices**, encuentra **un camino de longitud mínima** que los une



Algoritmo de Dijkstra: paso 0

- Separamos los vértices en 2 conjuntos, uno de ellos solamente tendrá a uno de los extremos con una distancia asociada (= 0)

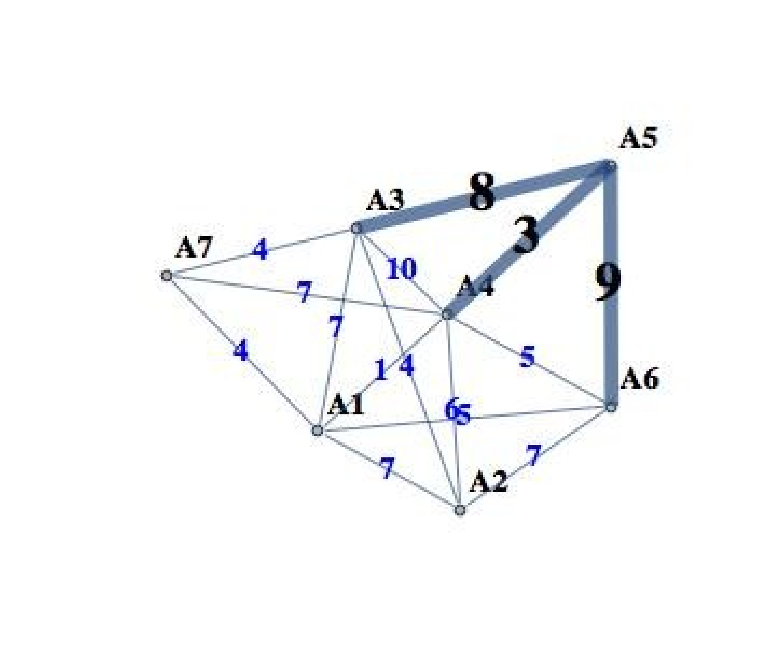
$V=\{(A_5,0)\}$ $W=\{A_1,A_2,A_3,A_4,A_5,A_7\}$



Algoritmo de Dijkstra: 1º paso

- Calculamos todas las ***distancias*** posibles entre vértices de **V** y vértices de **W**

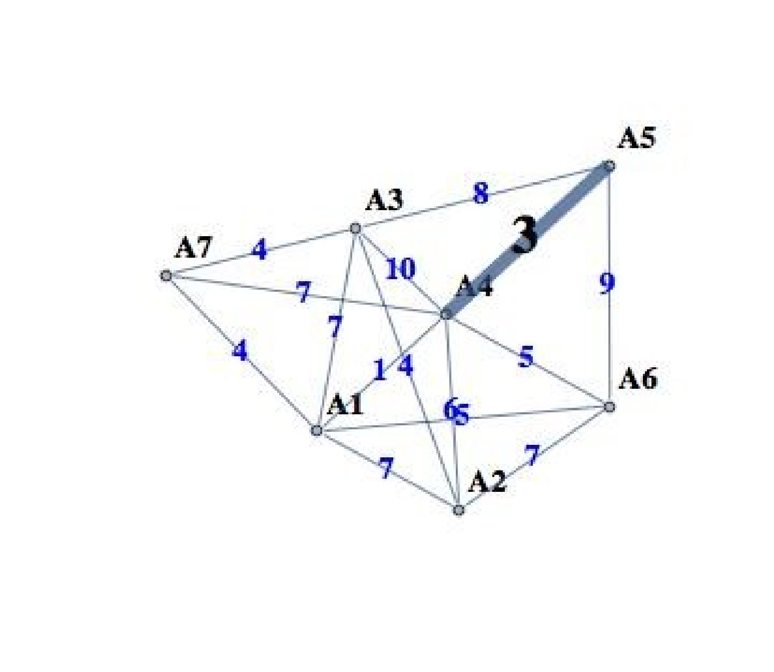
$V=\{(A_5,0)\}$ $W=\{A_1,A_2,A_3,A_4,A_6,A_7\}$



Algoritmo de Dijkstra: 2º paso

- Nos quedamos con *la* menor distancia, incorporamos ese vértice a **V** y lo quitamos de **W**

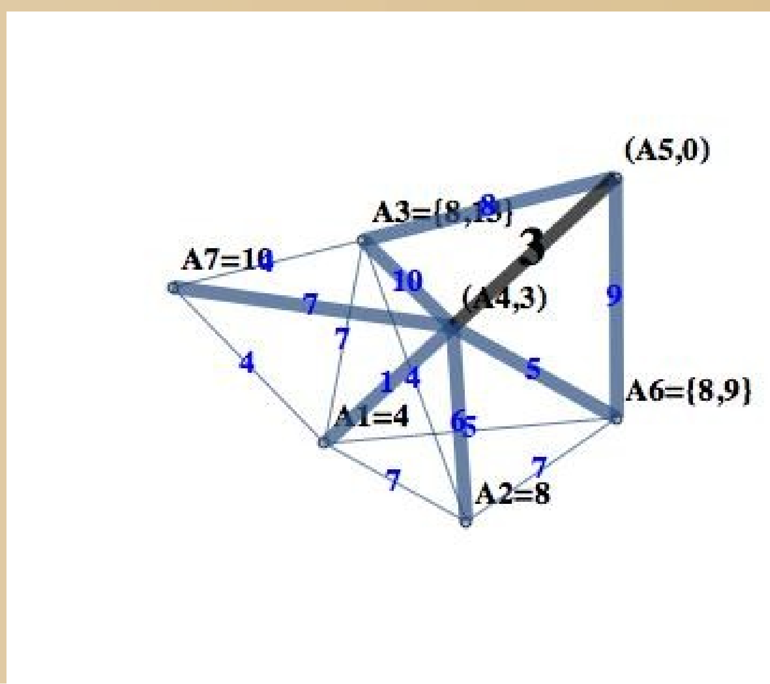
$V=\{(A_5,0),(A_4,3)\}$ $W=\{A_1,A_2,A_3,A_5,A_7\}$



Algoritmo de Dijkstra: 3º paso

- Repetimos el primer paso

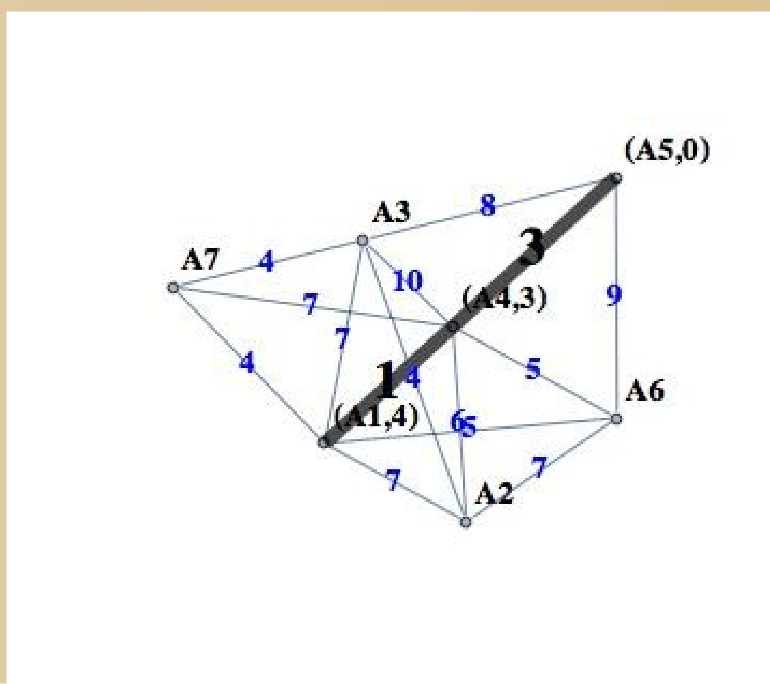
V={(A_5,0),(A_4,3)} **W={A_1,A_2,A_3,A_5,A_7}**



Algoritmo de Dijkstra: 4º paso

- Repetimos el segundo paso

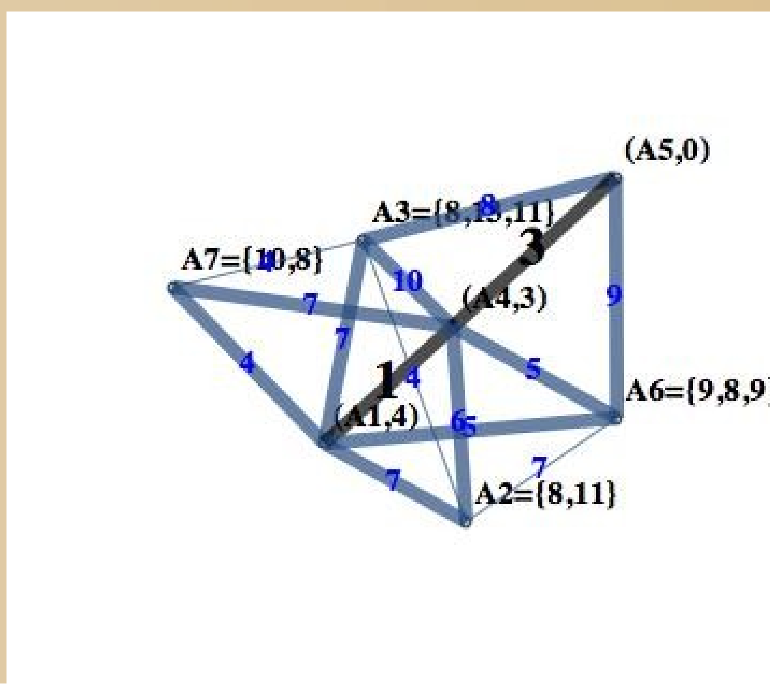
$V=\{(A_5,0),(A_4,3),(A_1,4)\}$ **$W=\{A_2,A_3,A_5,A_7\}$**



Algoritmo de Dijkstra: 5º paso

- Repetimos el primer paso

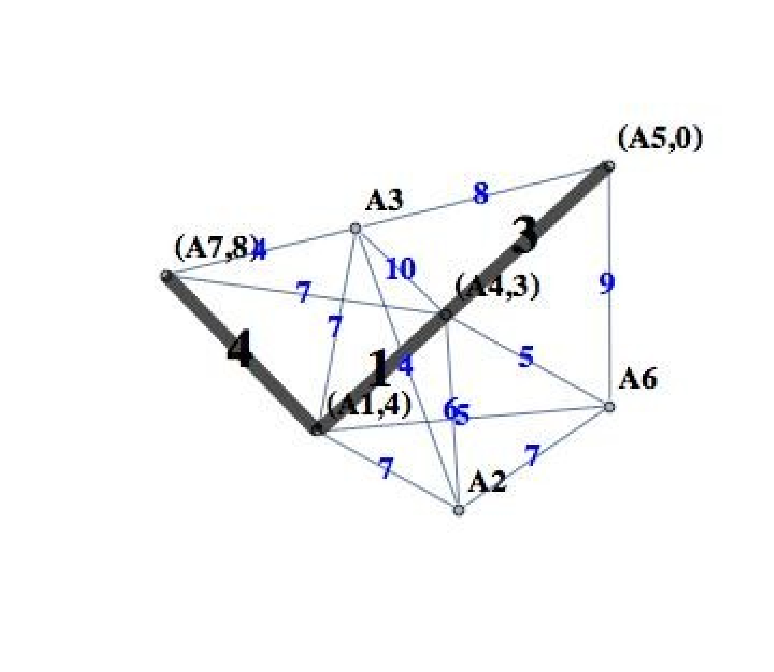
$V=\{(A_5,0),(A_4,3),(A_1,4)\}$ $W=\{A_2,A_3,A_5,A_7\}$



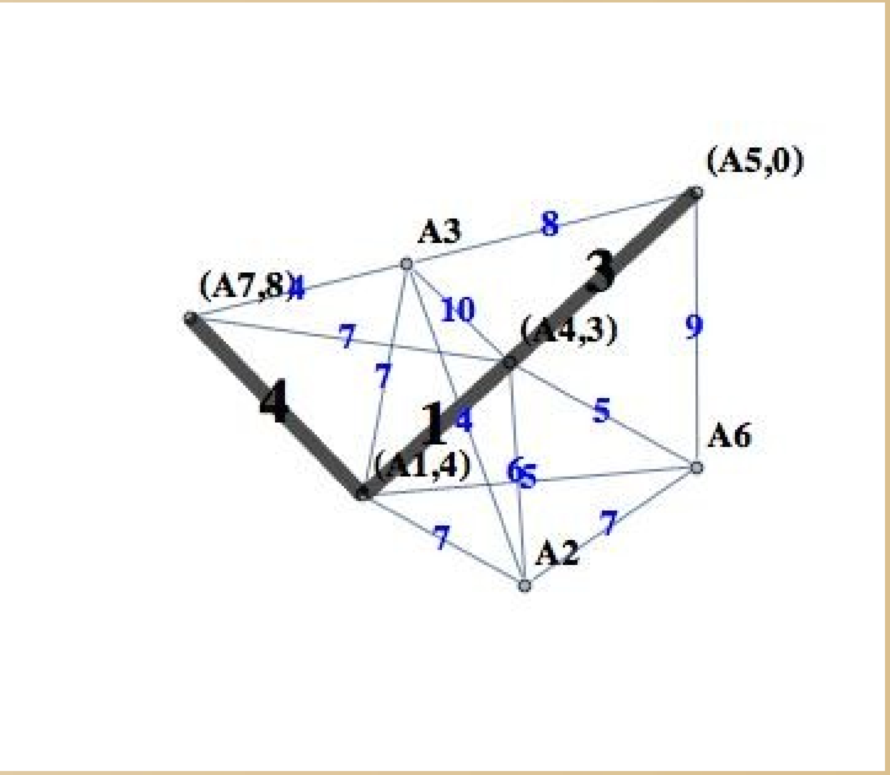
Algoritmo de Dijkstra: 6º paso

- Repetimos el segundo paso

$V=\{(A_5,0),(A_4,3),(A_1,4),(A_7,8)\}$ **$W=\{A_2,A_3,A_5\}$**

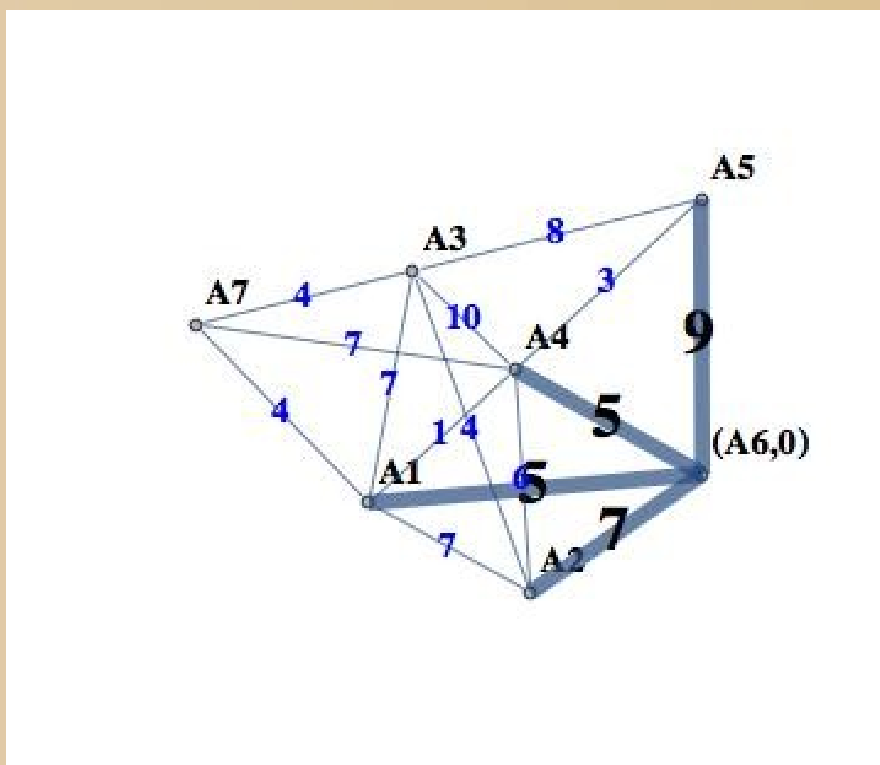


¡Problema resuelto!



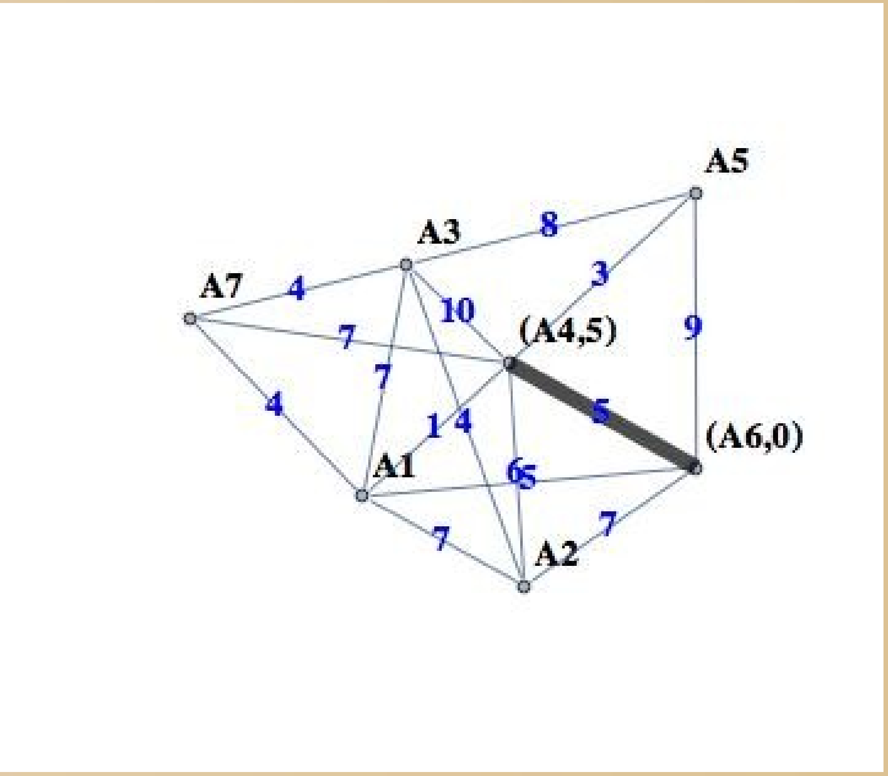
Camino más corto A_6- A_7

$V=\{(A_6,0)\}$ $W=\{A_1,A_2,A_3,A_4,A_5,A_7\}$



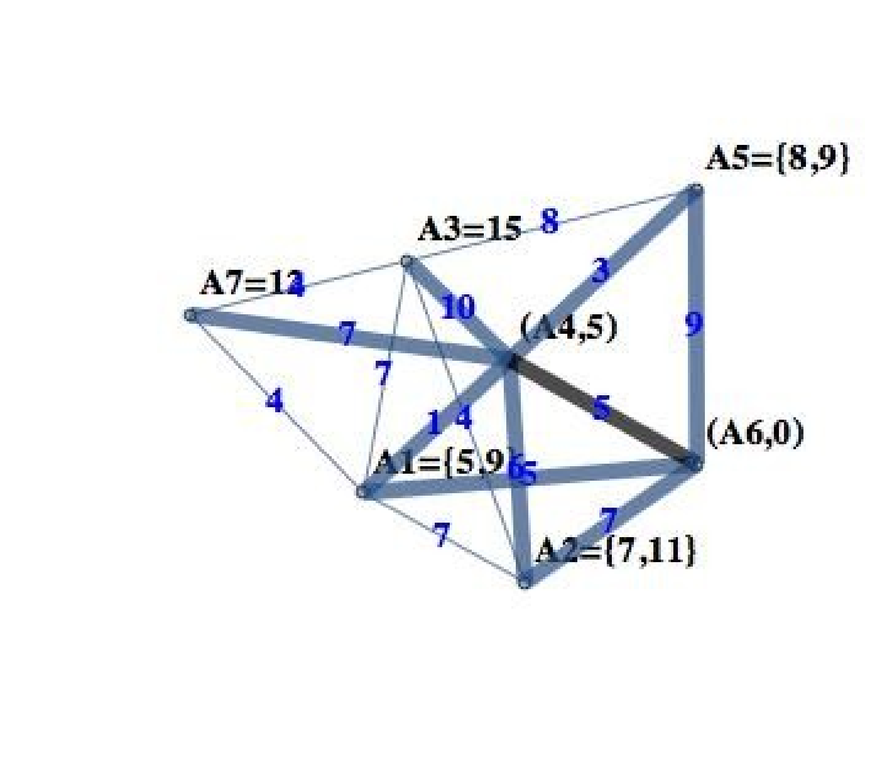
Camino más corto A_6- A_7

$V=\{(A_6,0),(A_4,5)\}$ **$W=\{A_1,A_2,A_3,A_5,A_7\}$**



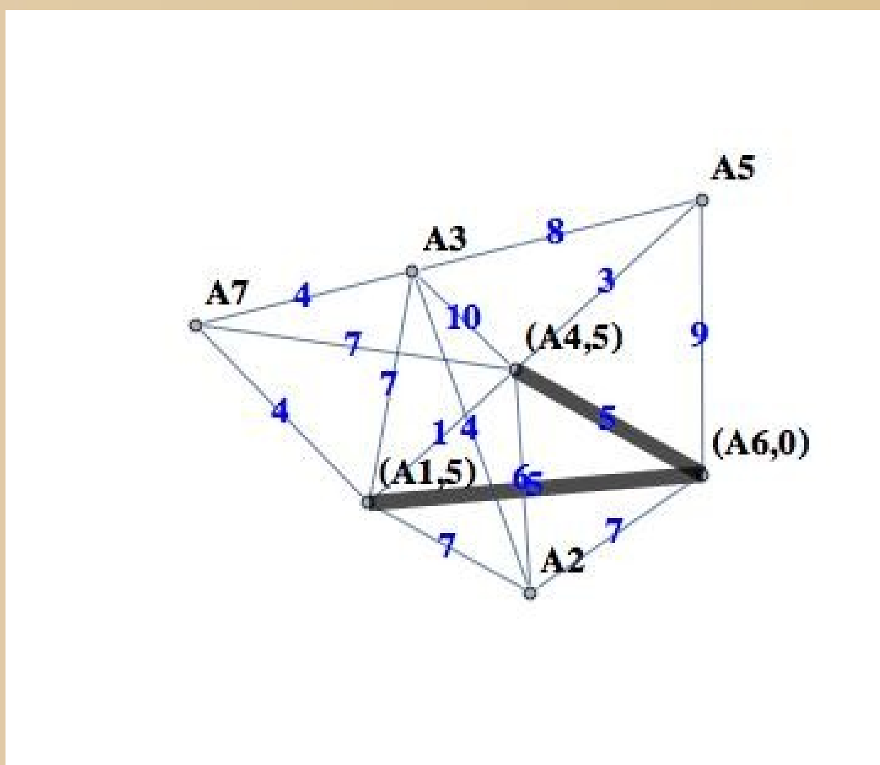
Camino más corto A_6- A_7

$V=\{(A_6,0),(A_4,5)\}$ $W=\{A_1,A_2,A_3,A_5,A_7\}$



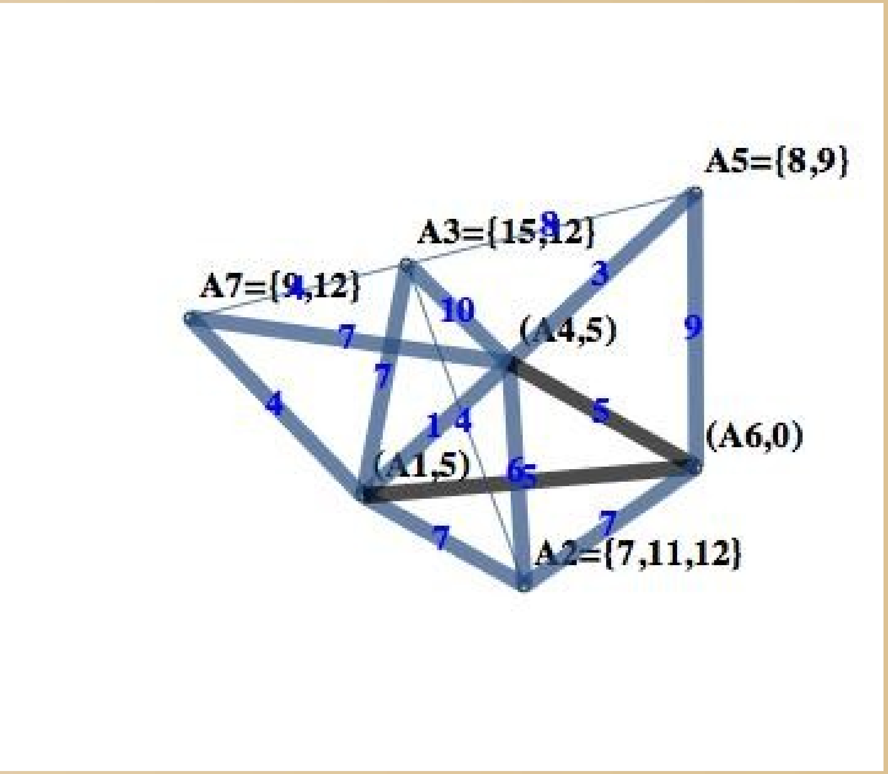
Camino más corto A_6- A_7

$V=\{(A_6,0),(A_4,5),(A_1,5)\}$ $W=\{A_2,A_3,A_5,A_7\}$



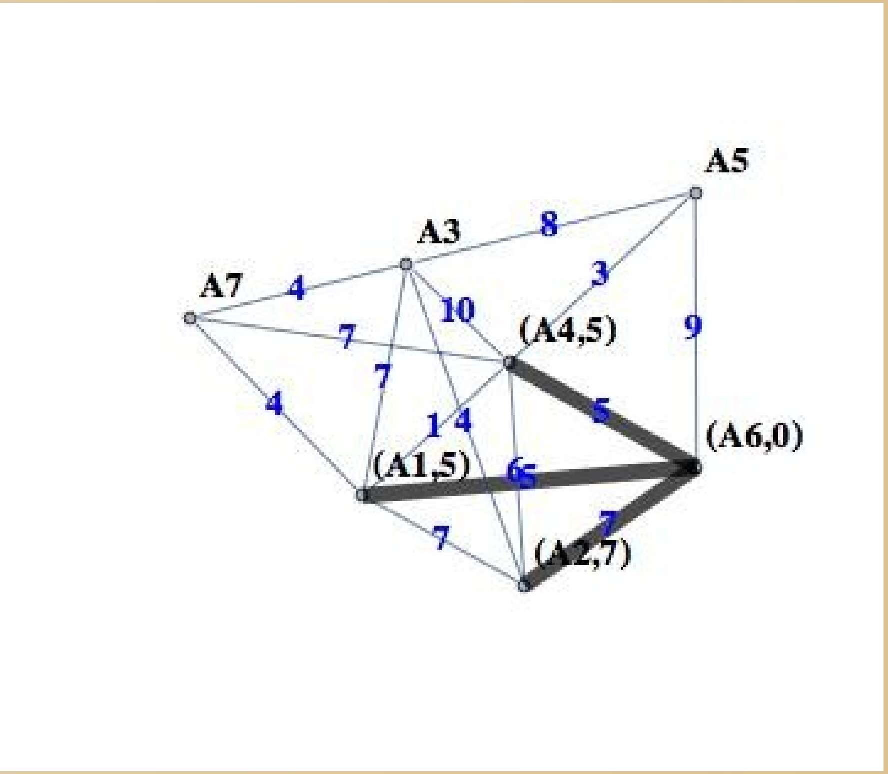
Camino más corto A_6- A_7

$V=\{(A_6,0),(A_4,5),(A_1,5)\}$ $W=\{A_2,A_3,A_5,A_7\}$



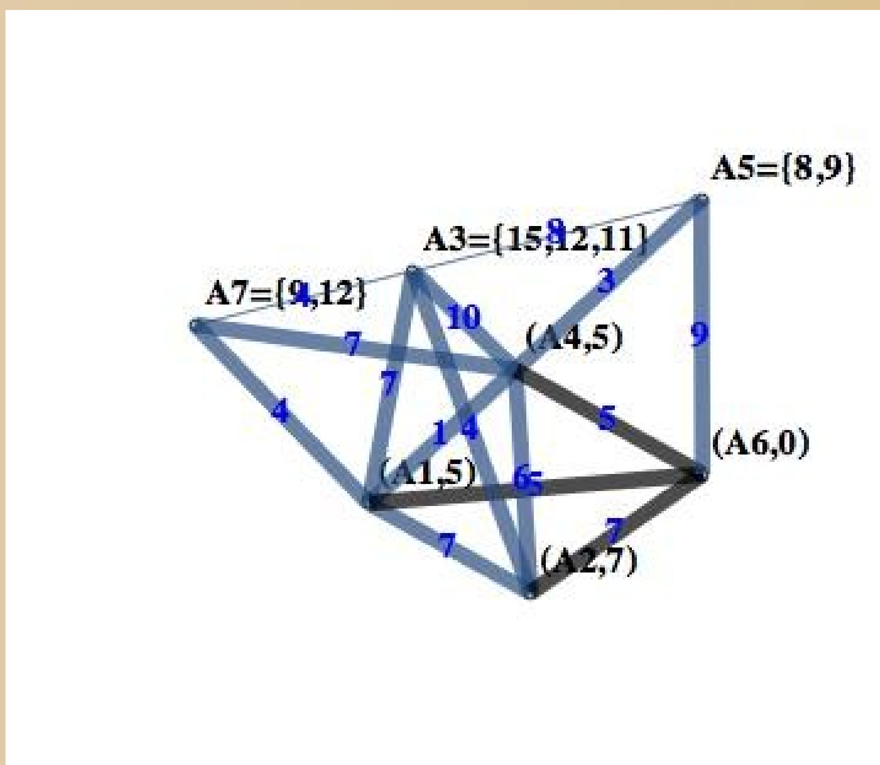
Camino más corto A_6- A_7

$V=\{(A_6,0),(A_4,5),(A_1,5),(A_2,7)\}$ $W=\{A_3,A_5,A_7\}$



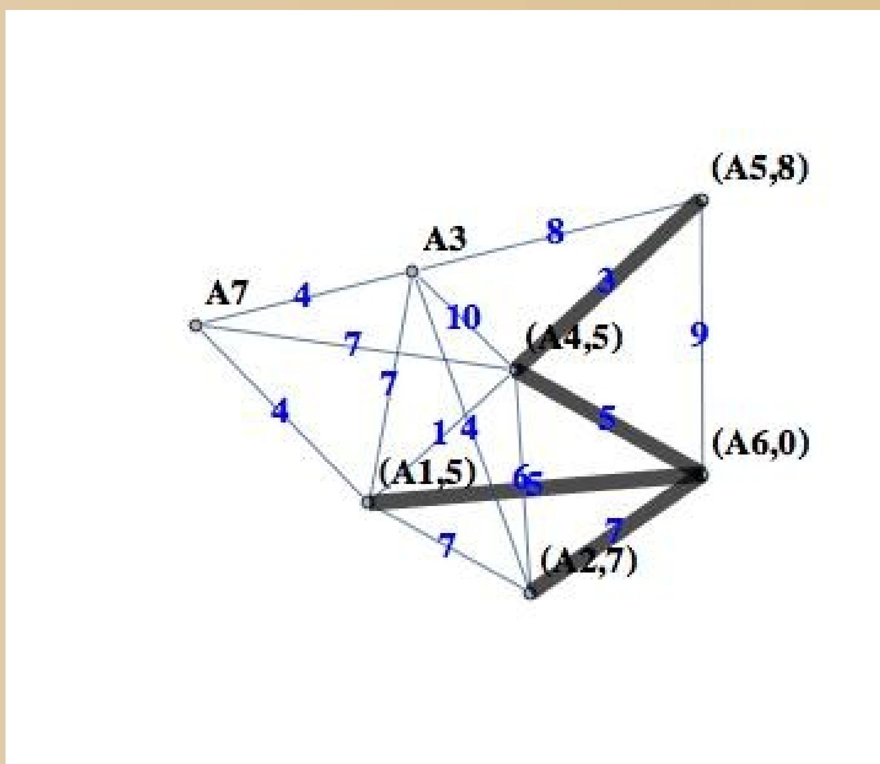
Camino más corto A_6- A_7

$V=\{(A_6,0),(A_4,5),(A_1,5),(A_2,7)\}$ $W=\{A_3,A_5,A_7\}$



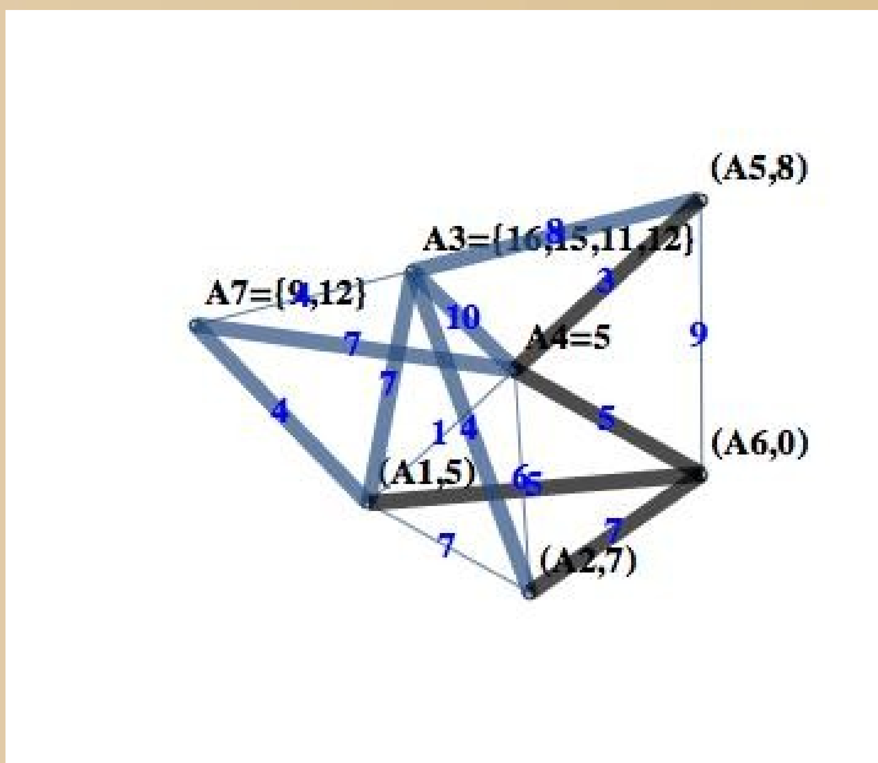
Camino más corto A_6- A_7

$V=\{(A_6,0),(A_4,5),(A_1,5),(A_2,7),(A_5,0)\}$ **$W=\{A_3,A_7\}$**



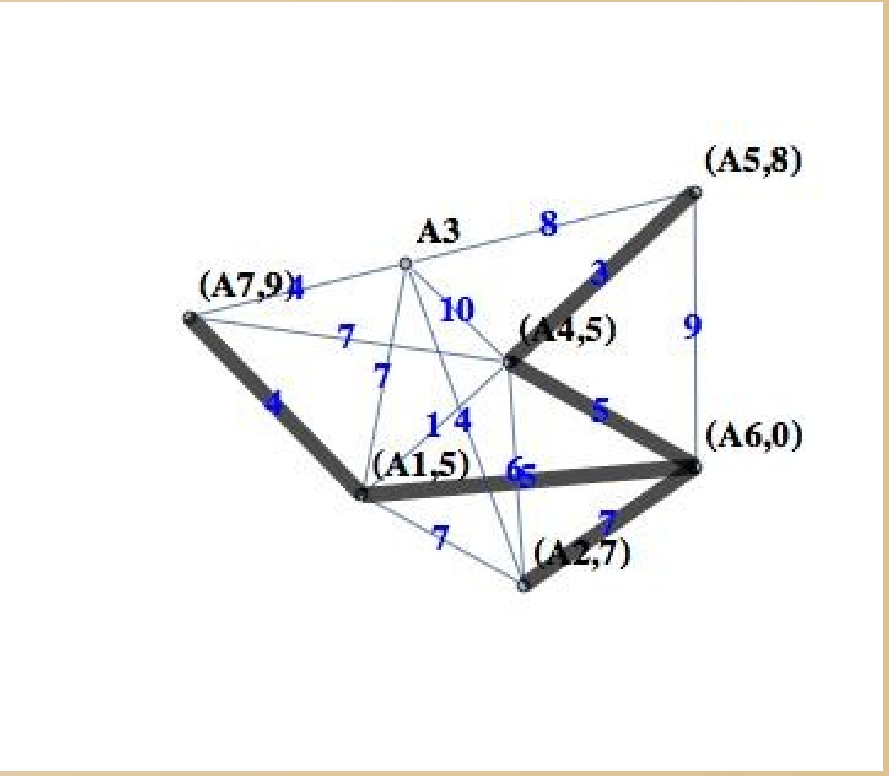
Camino más corto A_6- A_7

$V=\{(A_6,0),(A_4,5),(A_1,5),(A_2,7),(A_5,0)\}$ **$W=\{A_3,A_7\}$**



Camino más corto A_6- A_7

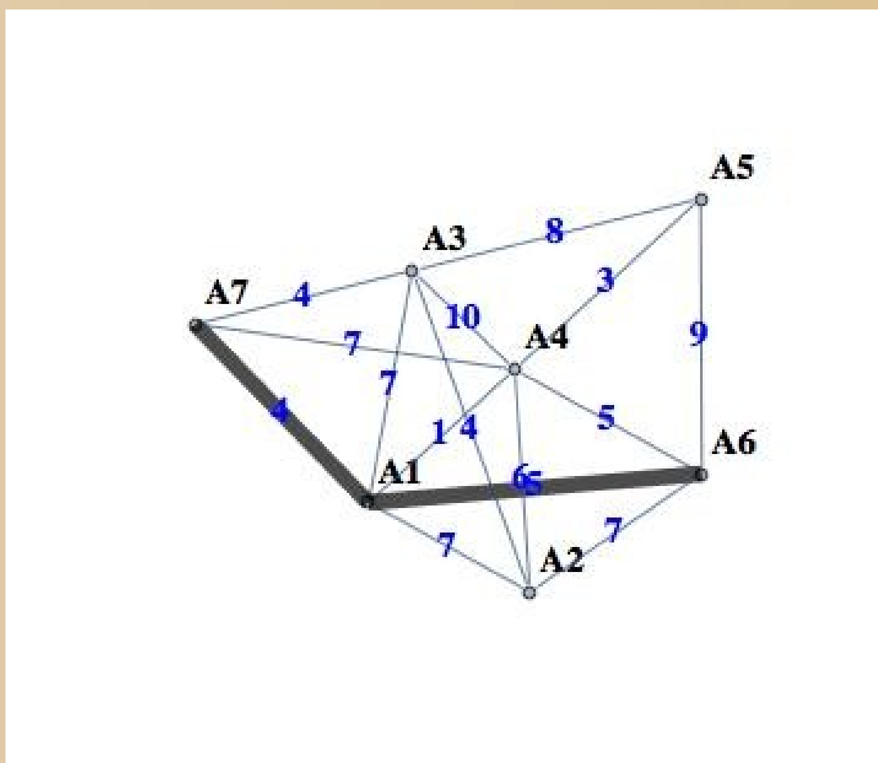
V={(A_6,0),(A_4,5),(A_1,5),(A_2,7),(A_5,8)(A_7,9)} **W={A_3}**



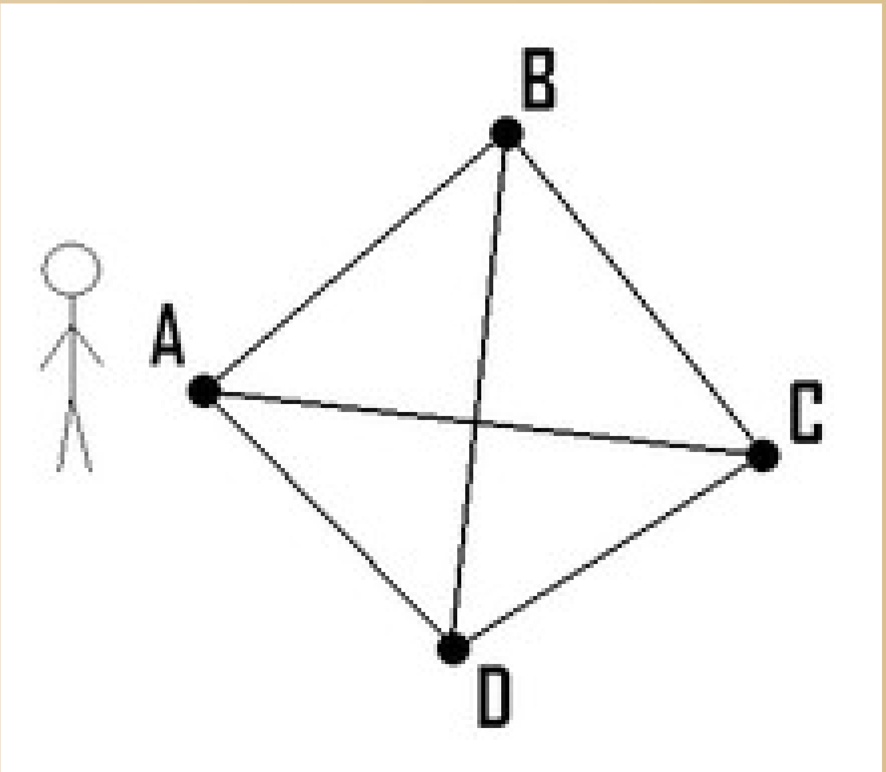
Y ***el*** camino más corto es...

$$V=\{(A_6,0),(A_4,5),(A_1,5),(A_2,7),(A_5,8)(A_7,9)\}$$

$$W=\{A_3\}$$

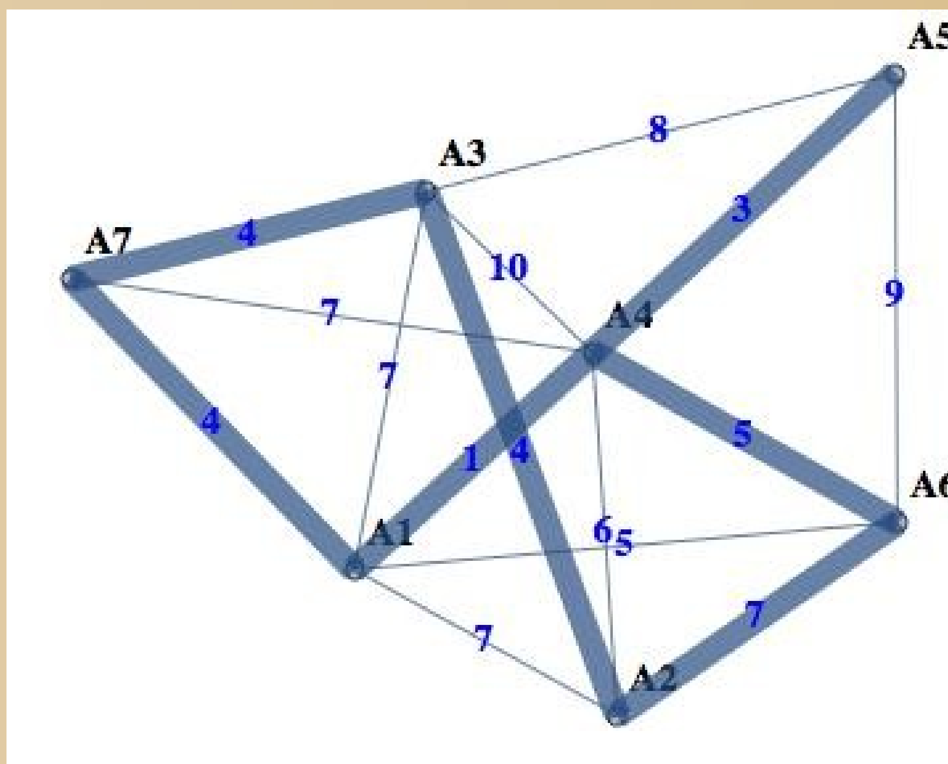


El Problema del Viajante (????)



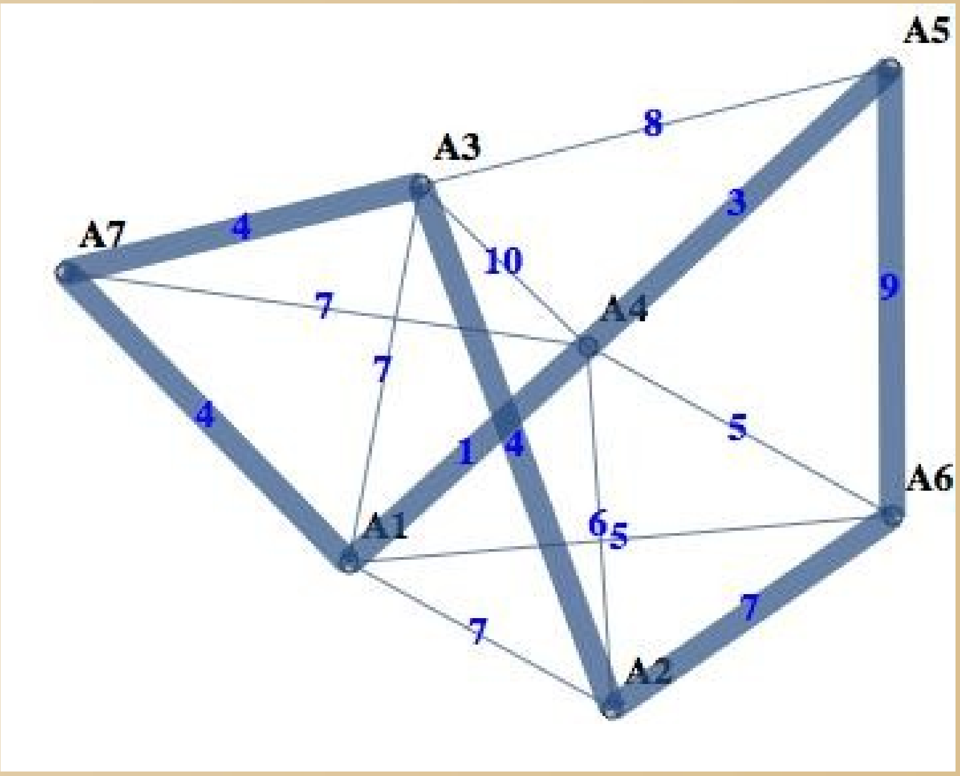
Y el paseo más corto es...

$$A_1 \rightarrow A_7 \rightarrow A_3 \rightarrow A_2 \rightarrow A_6 \rightarrow A_4 \rightarrow A_5 \rightarrow A_4 \rightarrow A_1 = 31$$

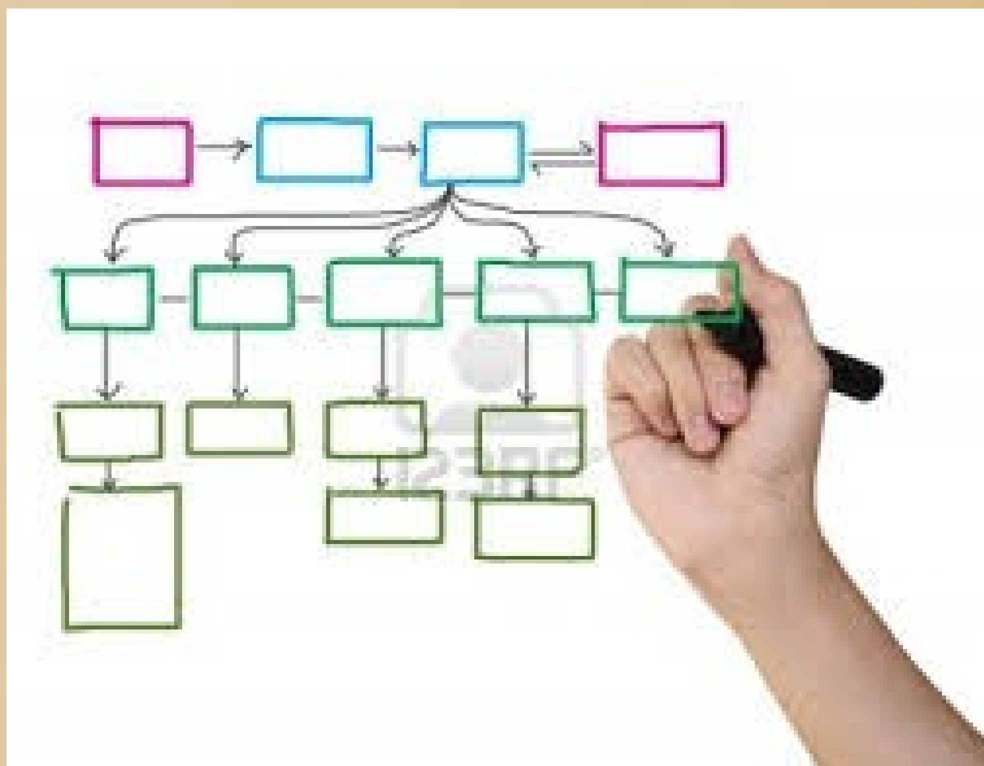


Sin pasar 2 veces por el mismo lugar..

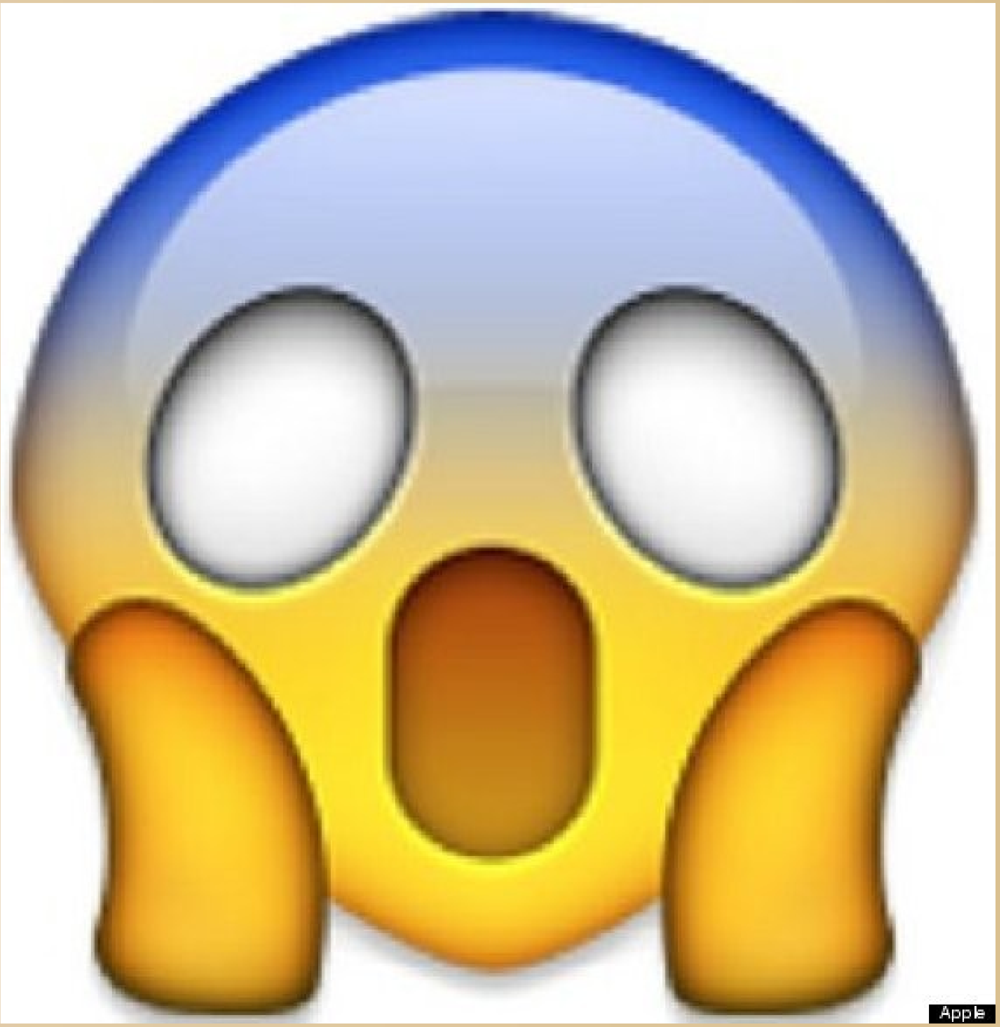
$$A_1 \rightarrow A_7 \rightarrow A_3 \rightarrow A_2 \rightarrow A_6 \rightarrow A_5 \rightarrow A_1 = 32$$



¿Algoritmo eficiente???



¡No hay!



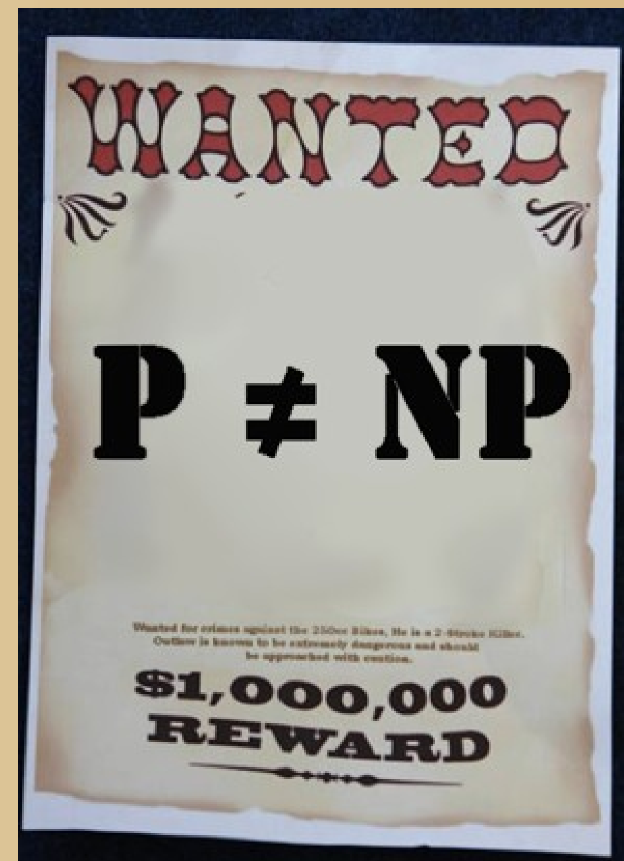
Mejor dicho...

Mejor dicho...

- **No se sabe si hay un algoritmo o no**

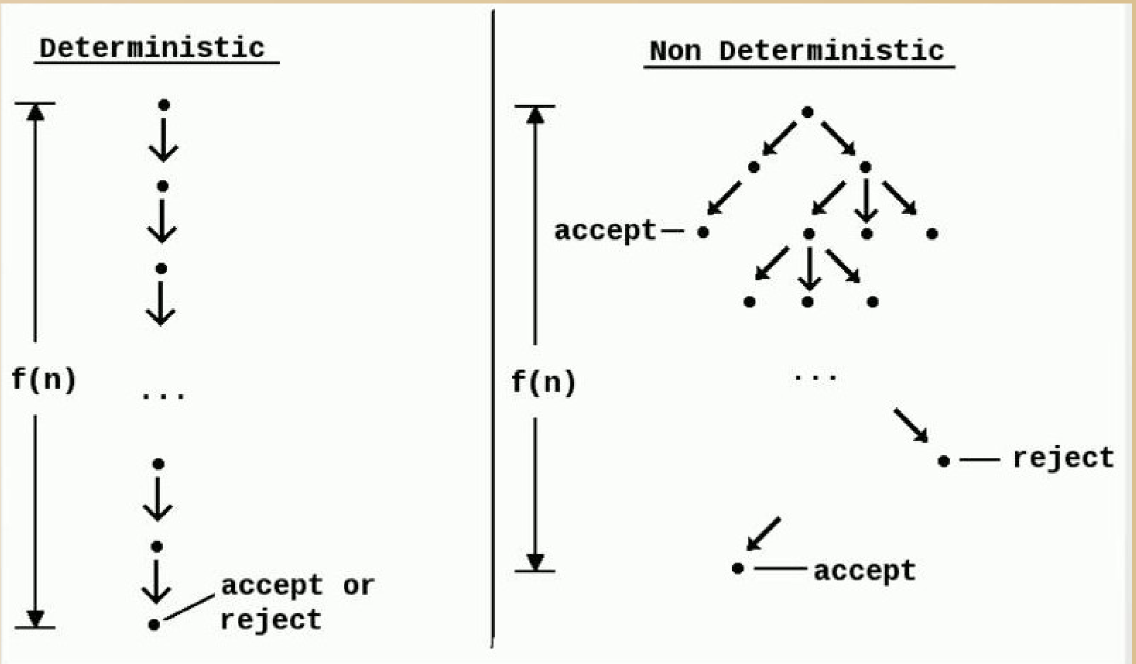
Mejor dicho...

- **No se sabe si hay un algoritmo o no**
- **Desde el año 2000 hay una recompensa de 1.000.000 U$D para quien "*resuelva*" el problema**



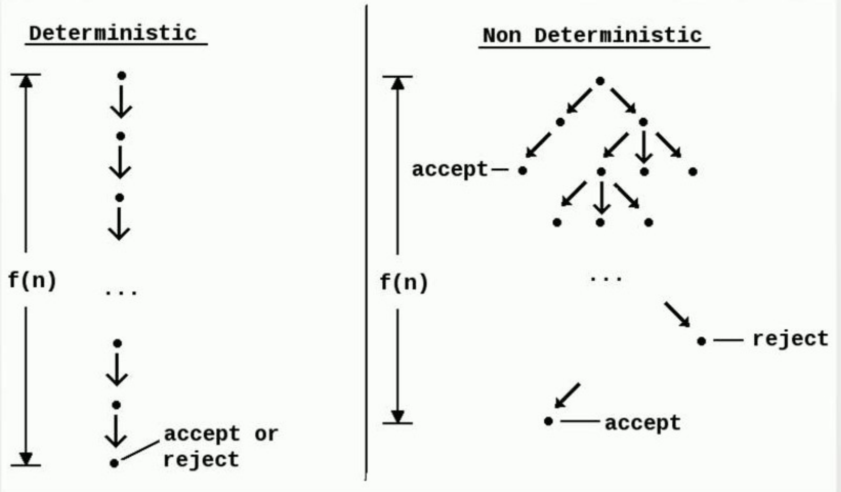
CLAY
MATHEMATICS
INSTITUTE

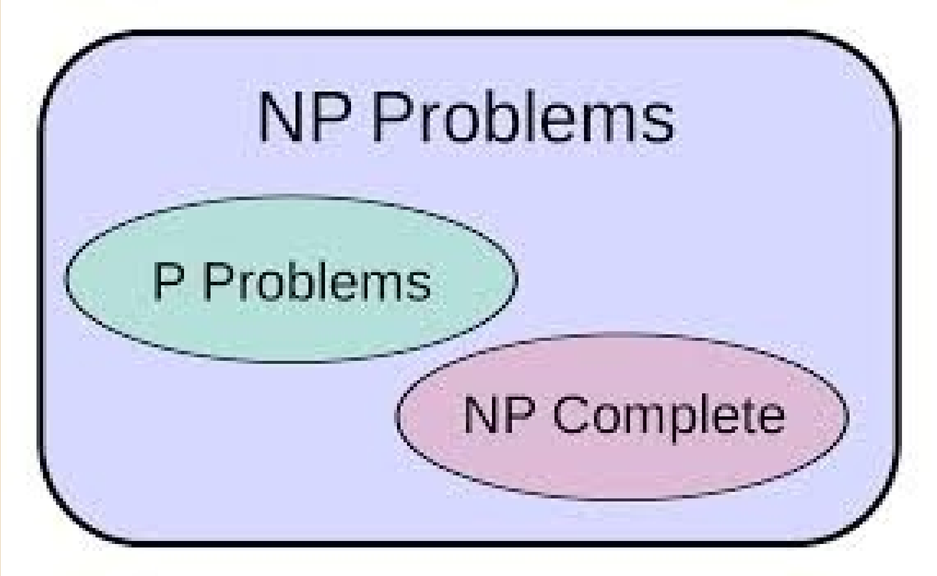
P vs NP?



P = Polinomial (determinístico)
NP = Polinomial (no determinístico)

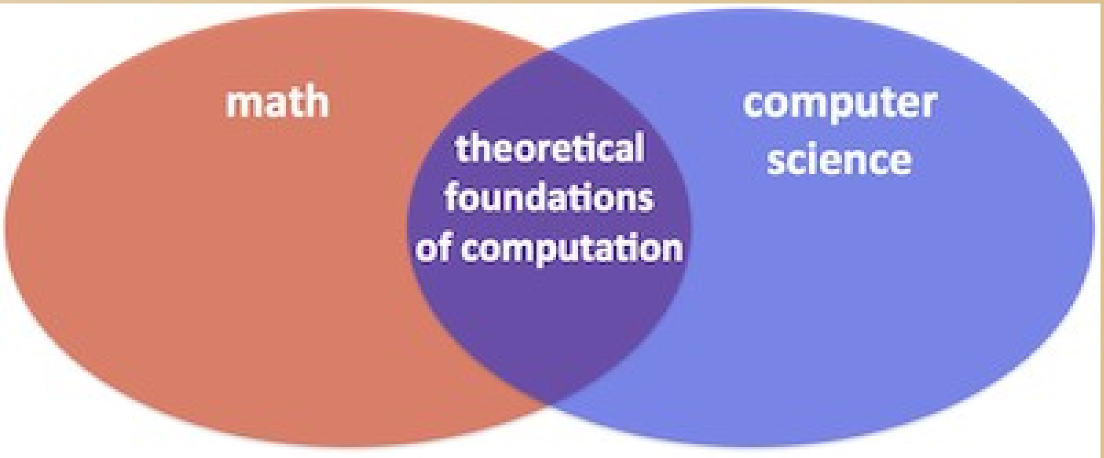
P vs NP


Deterministic
f(n)
accept or reject
Non Deterministic
f(n)
accept
reject
accept


NP Problems
P Problems
NP Complete

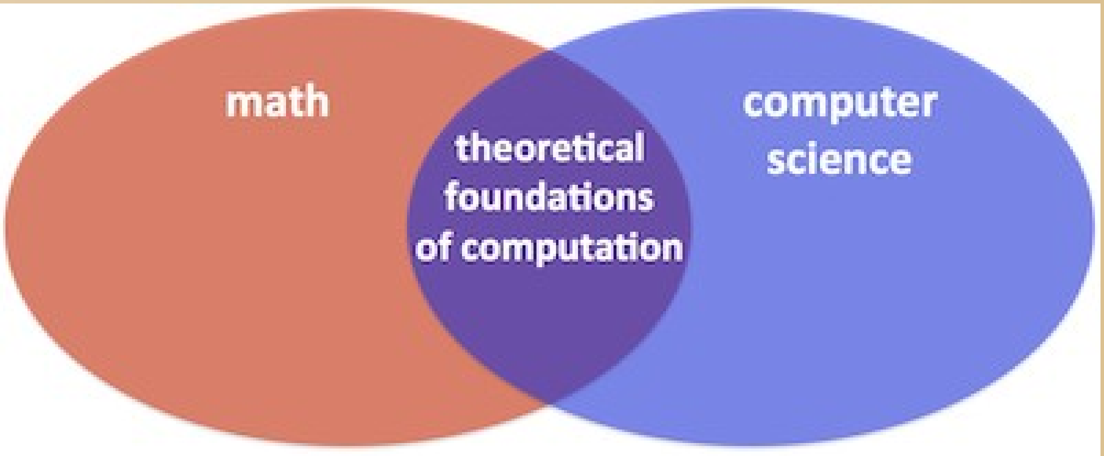

257
Date
Pay to the order of
Cash
$ 1.000.000
One Million
Dollars
BANK
BANK
985736592
8462027561
0257
dreamstime.com

Y si os va este rollo...



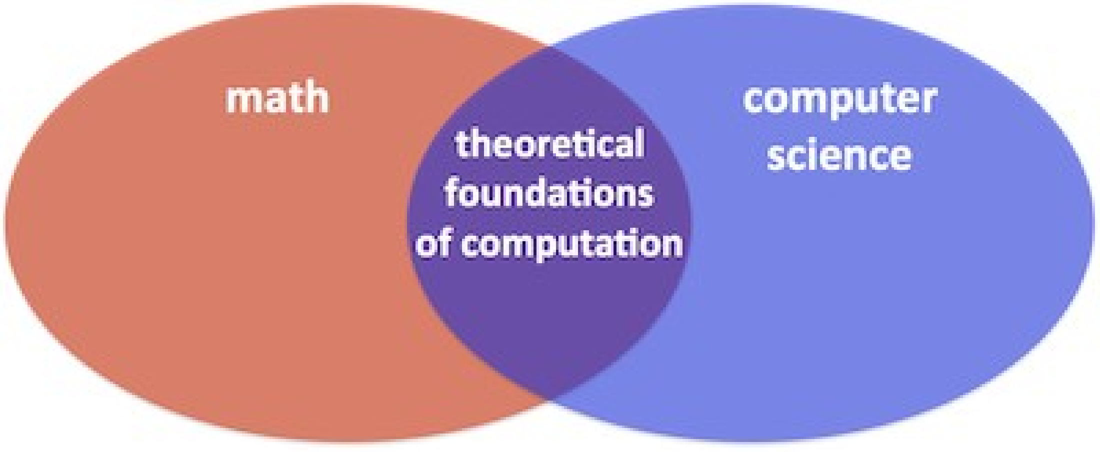
Y si os va este rollo...

- **Matemática Discreta**



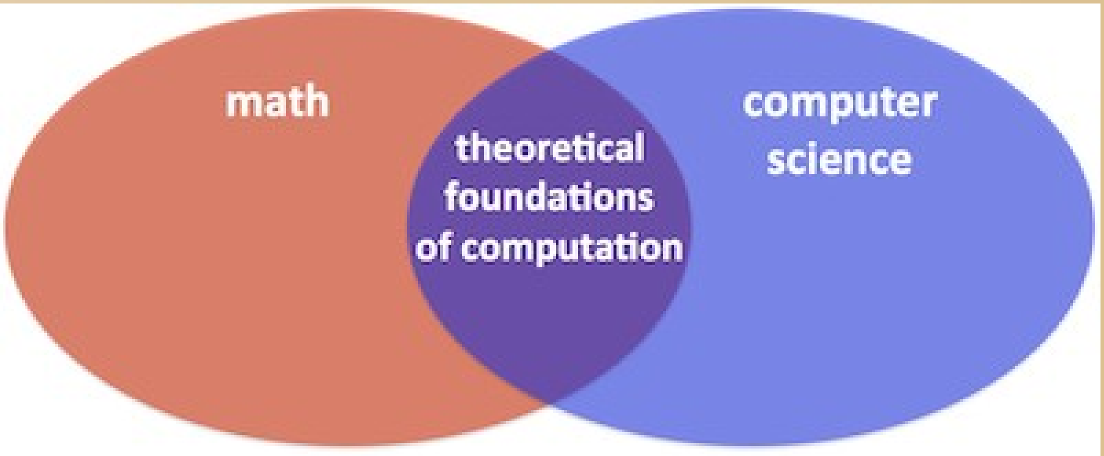
Y si os va este rollo...

- **Matemática Discreta**
- **Lógica**



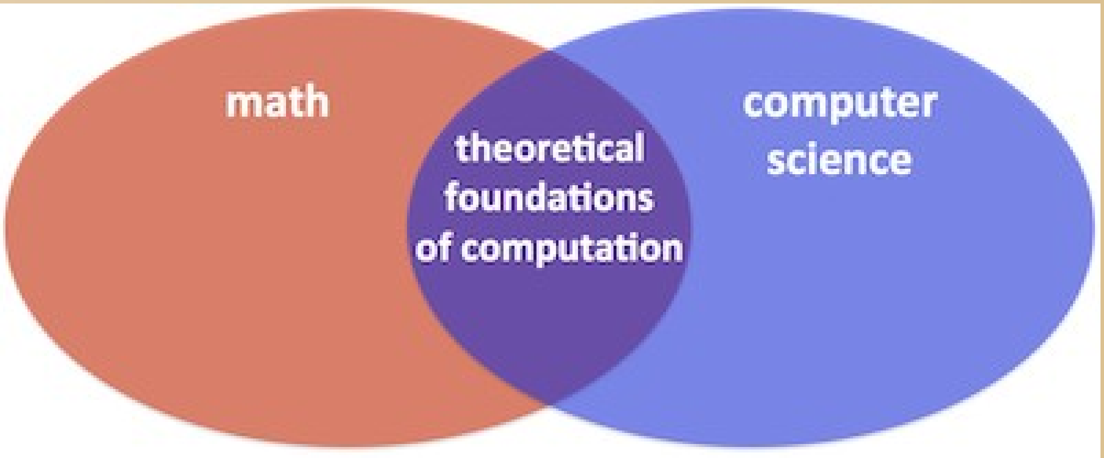
Y si os va este rollo...

- **Matemática Discreta**
- **Lógica**
- **Algoritmos**



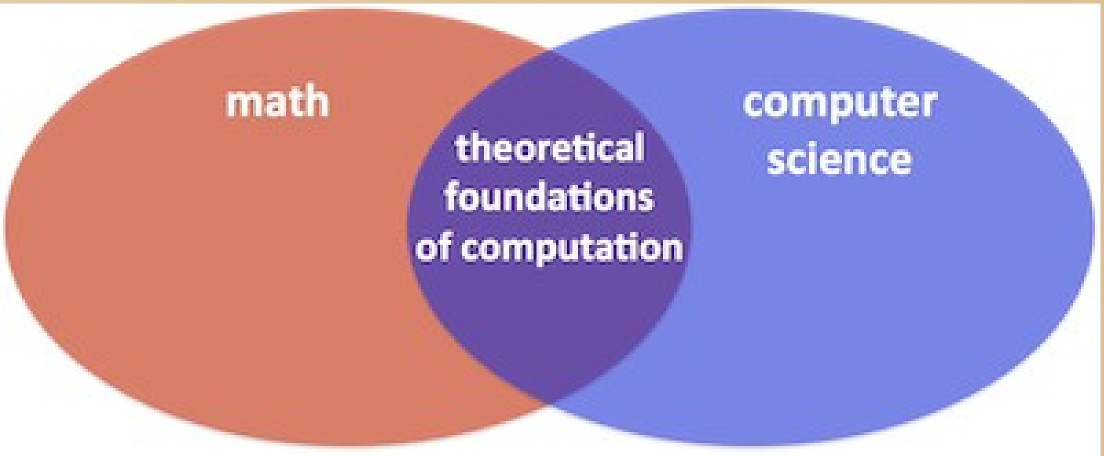
Y si os va este rollo...

- Matemática Discreta
- Lógica
- Algoritmos
- Programación



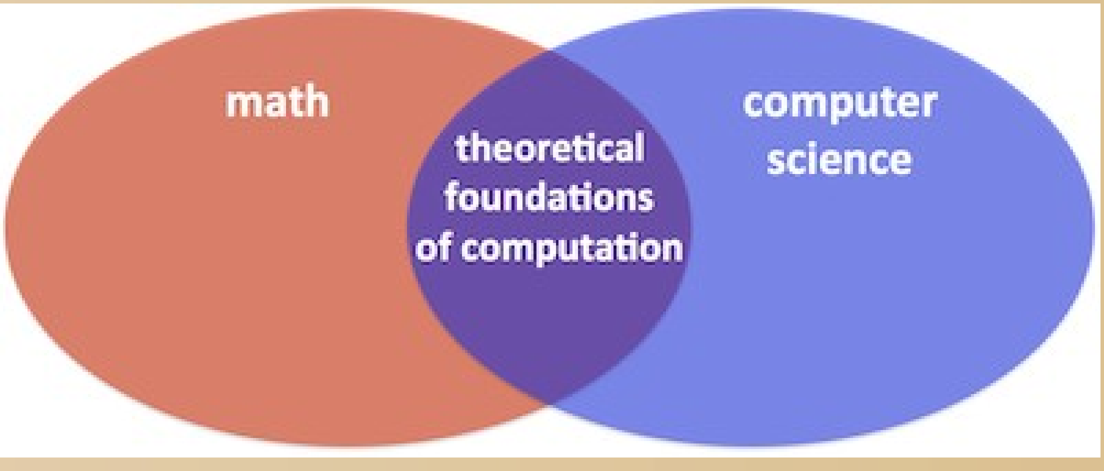
Y si os va este rollo...

- Matemática Discreta
- Lógica
- Algoritmos
- Programación
- Complejidad



Y si os va este rollo...

- Matemática Discreta
- Lógica
- Algoritmos
- Programación
- Complejidad
- ...




A
B
C
D


"That's all Folks!"